

# **Automated Supermarket Report 1**

Group 4

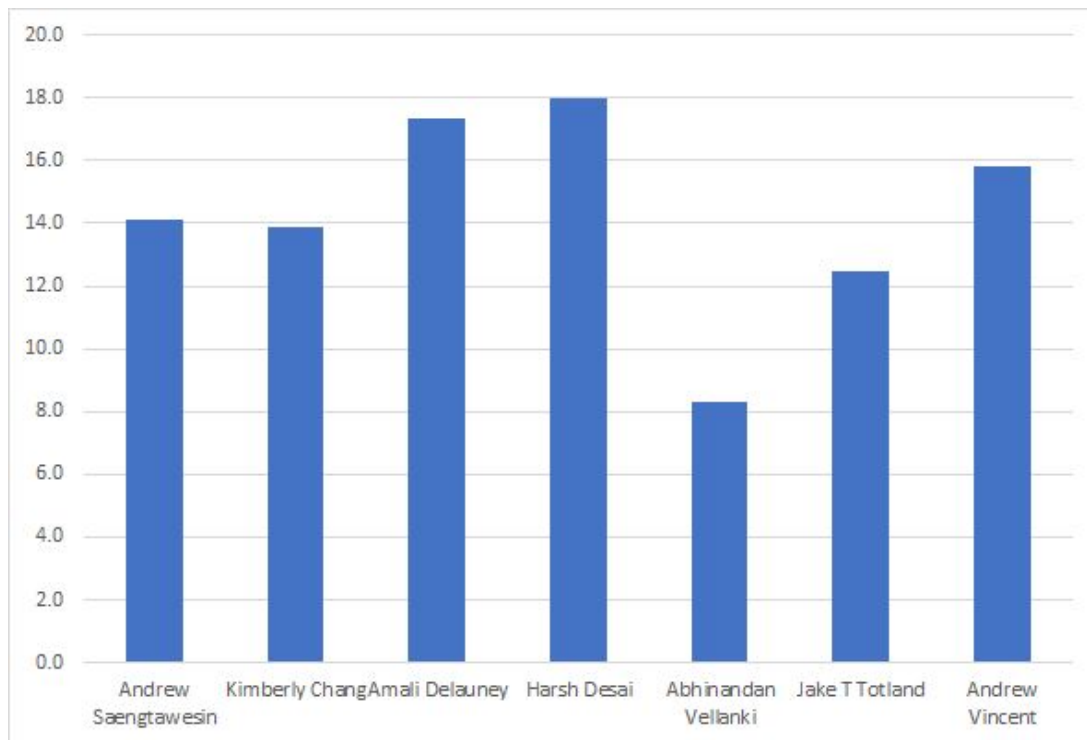
[github.com/as2580/SoftwareEngineeringGroup4](https://github.com/as2580/SoftwareEngineeringGroup4)

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Section 0: Individual Contributions Breakdown</b>	<b>2</b>
<b>Section 1: Customer Problem Statement</b>	<b>3</b>
Section 1.1: Problem Statement	3
Section 1.2: Glossary of Terms	7
<b>Section 2: System Requirements</b>	<b>8</b>
Section 2.1: Enumerated Functional Requirements	8
Section 2.2: Enumerated Nonfunctional Requirements	11
Section 2.3: User Interface Requirements	13
<b>Section 3: Functional Requirements Specification</b>	<b>17</b>
Section 3.1: Stakeholders	17
Section 3.2: Actors and Goals	17
Section 3.3.1: Casual Description	18
Section 3.3.2: Use Case Diagrams	20
Section 3.3.3: Traceability Matrix	22
Section 3.3.4: Fully-Dressed Descriptions	25
Section 3.4: System Sequence Diagrams	30
<b>Section 4: User Interface Specification</b>	<b>35</b>
Section 4.1: Preliminary Design	35
Section 4.2: User Effort Estimation	42
<b>Section 5: Domain Analysis</b>	<b>44</b>
Section 5.1: Domain Model	44
Section 5.1.1: Concept Definitions	44
Section 5.1.2: Association Definitions	45
Section 5.1.3: Attribute Definitions	46
Section 5.1.4: Traceability Matrix	48
Section 5.1.5: Application Domain Model	49
Section 5.2: System Operation Contracts	50
<b>Section 6: Project Size Estimation Based on Use Case Points</b>	<b>51</b>
<b>Section 7: Plan of Work</b>	<b>52</b>
<b>Section 8: References</b>	<b>56</b>

## Section 0: Individual Contributions Breakdown

	Points	Andrew Saengtawesin	Kimberly Chang	Amali Delauney	Harsh Desai	Abhinandan Vellanki	Jake T Totland	Andrew Vincent
Project Management	10	60%	40%					
Sec. 1: Customer Problem Statement	9	20%	80%					
Sec. 2: System Requirements	6	80%	20%					
Sec. 3: Functional Requirements Specification	30	5%	5%	30%	60%			
Sec. 4: User Interface Specs	15						50%	50%
Sec. 5: Domain Analysis	25			33%		33%		33%
Sec. 6: Plan of Work	5						100%	
Total Points	100	14.1	13.9	17.3	18	8.3	12.5	15.8



# **Section 1: Customer Problem Statement**

## **Section 1.1: Problem Statement**

Supermarkets can be chaotic places with all the customers scurrying around looking for their items and workers trying to keep stock on shelves. A disorganized and poorly managed supermarket can alienate customers and cause business to move elsewhere. In the supermarket industry, ensuring customer satisfaction is of utmost importance.

In the current market, adaptation to the usage of new technologies is essential. Large companies such as Walmart and Amazon are becoming the dominant players in the industry, so supermarkets must have competing technologies. The supermarket Earth Flare has already been forced out of business by these behemoths. [\[1\]](#) Walmart has a pick up in store option for customers, and Amazon has its own physical store Amazon Go automated without any employees.

It is imperative to implement technologies in a supermarket to increase the convenience of the shopping experience to better ensure customer satisfaction and retention, increase worker productivity, and reduce redundancy and inefficiencies.

### **Problem 1: Item Management**

Often, customers will be unable to find a specific item and must then search the store to find an employee to help them find the item. Shelves may be left empty until a customer asks an employee if there is any stock in the backroom. Items may simply be out of stock in the store, and more must be ordered from the supplier. Being unable to locate items could cause great dissatisfaction in customers, so managing stock is important for the maintenance of a supermarket.

Often, customers will place items on the wrong shelf because doing so is more convenient than returning the item to the proper shelf. This can cause disorganization of the inventory on the shelves, make finding items much harder for other customers, and create unnecessary tasks that must be handled by the employees. Therefore, making the process of removing unwanted items from a shopping cart should be made more convenient.

Expired items may be left on shelves, causing serious potential liabilities. Customers should not ever be able to add expired items to their shopping cart. The potential lawsuits of such a small oversight could cost a great amount for the supermarket.

## **Solution**

Each product shall have an RFID, and each item will have its own RFID tag to allow for easier item scanning. Entire stores can be run without employees using items with RFID tags.<sup>[2]</sup>

At the end of each aisle, there will be a Shopping Assistance Terminal. The Shopping Assistance Terminal will be attached to the wall instead of a price checker; it will not be removable by the customer in order to prevent theft. The Shopping Assistance Terminal will run the Customer Application to allow the customer to (1) check the location(s) of an item or items for which they are searching or (2) check the price of an item. When checking the location of an item, the customer will be shown a map of the store, and the aisle (ex. Aisle 7) or area (Produce) of the store in which the item is located will be highlighted. The Customer Application will also be available to download on the customer's smart device in order to improve convenience and encourage customer retention.

There will be a basket for customers to leave an item to be reshelfed. The RFID scanner inside the basket will constantly check to see if items are in the basket, and if an item is inside, a task will be generated in the Employee Application, and an alert will be sent to employees. This will improve the convenience of removing unwanted items from shopping carts and improve store item organization.

All items shall be contained in a database. There shall be a separate inventory database for the items on shelves and for the items in the back room. Each entry in the inventory database shall contain a product's name, the number of that product that are on the shelves, the location of that product, its RFID, and its expiration date (if applicable). The back room database shall have the same categories, but will keep track of the items in the backroom. For items with expiration date, they will be group in bundles (as most products are bought in bulk), and each bundle of items will be grouped together with the earliest expiration date.

## **Problem 2: Task Synchronization**

Multiple employees may be performing the same task, resulting in redundancy and inefficiency. This could cause tasks to pile up and force employees to complete the plethora of tasks that need to be completed. Employees may also find it difficult to determine what tasks need to be done without manually monitoring the store to discover that an issue needs to be resolved.

Efficiency is key to running a supermarket smoothly. Employee tasks should be synchronized and organized in order to allow employees to perform their work more efficiently. Disorganized employees will not be able to help customers smoothly and may cause the supermarket to appear substandard.

### **Solution**

There shall be an Employee Application that will create tasks that employees can accept and complete. Tasks will be created for when a customer has placed an item to be reshelfed in the Shopping Assistance Terminal basket, when a customer has requested assistance on the Customer Application, when items have past their expiration date and should be thrown out, and when items need to be restocked (in either the store or the back room). An employee can claim a task, but multiple employees cannot claim the same task. When an employee claims a task, the task will be marked as having been assigned. When an employee completes a task, they will mark the task as complete, and the task will be removed.

Employees will be able to clock in and out on the Employee Application to determine employee hours more easily. Managers will be able to view and correct employee hours on the Manager Application. If an employee requires assistance, they will be able to request help from a manager.

### **Problem 3: Self-Checkout and Returns**

Many supermarkets have a cashier to manually checkout customers. This can be inefficient because at times, a cashier may get no customers, and they will not have any work to do. Moreover, the speed of a customer's checkout relies on the speed of the cashier, and if the cashier is slow, this could cause an irritating and unpleasant experience for the customer, especially if they have been waiting in a line for a long time. Self-checkouts can help with this issue. Although some supermarkets have self-checkouts, they are still moderated by employees; it would be better if this system could be completely automated.

Returns can also be an irritating situation, and the process should be better automated. The process should still be moderated by an employee because a customer could potentially return an empty box, bag, etc. that may be detected by video as being a returnable item. However, expediting this process will improve efficiency and improve the customer experience.

### **Solution**

Checkouts will be completely automated. There shall be Checkout Terminals by the exit. A customer will be able to use the Checkout Terminal to checkout their items. A customer can bag several items and leave them in the bagging area. The customer can then use the application on the Checkout Terminal to determine which items were bagged. The customer may also remove that set of items from the bagging area and begin the process with the next set of items. If a customer requires assistance, they can request it on the application on the Checkout Terminal.

There will be a basket for items that a customer decides that they do not want. This will allow customers to easily remove unwanted items from their shopping cart without buying the item and without leaving the unbought item at the Checkout Area with employees unaware of the issue.

There will be a Returns Terminal where customers can return items that have already been purchased. The station will be run by an employee to ensure proper returns are made. The employee will be able to scan the customer's receipt to ensure it contains the item to be returned. There shall be an RFID scanning basket at the terminal that can determine the item to be returned. The employee running the Returns Terminal can then create a task for those items to be reshelfed.

#### **Problem 4: Management**

Supermarkets will always have employee turnaround. Many low-level employees only work there temporarily in preparation for other opportunities. Therefore, training should be minimized so as to improve efficiency.

Moreover, training excellent managers can be a difficult and time-consuming task, which includes finding the right people and providing enough training to allow them to reach their full potential in a managerial position. There will be many more low-level employees than there will be managers, and even the best managers can be overwhelmed by having to supervise so many workers. Furthermore, managers will have to retire eventually, and a lot of time and effort must be spent to replace them.

#### **Solution**

The proposed system will be simple for employees to use, and since the learning curve for this system is so low, the training time could be reduced from a few hours to just a few minutes. Rather than requiring a manager to micromanage each employee directly, employees can be given clear instructions on what needs to be done through the Employee Application. This will also augment a manager's ability to supervise employees. With this system, employee turnaround will not be an issue because the system will be simple to learn.

#### **Costs**

This project will require tablets to be installed in each aisle, RFID tags for each item in the store, and RFID readers for checkout and returns.

Android tablets may be bought for under \$50 each when bought in bulk. 10 tablets should suffice to be used in the Shopping Assistance Terminal. 6 tablets should suffice for use in the Checkout Terminal. 1 tablet should suffice for use in the Returns Terminal. The tablets would cost approximately \$850.

RFID tags can be bought for \$0.05 or less when bought in bulk. Supermarkets hold around 15,000 to 60,000 different products.<sup>[3]</sup> At least 100 of each product are in stock. The example supermarket shall contain 200,000 items. The RFID tags would cost approximately \$10,000; however, this price can be offset by increasing the price of products by a small amount.

Smaller RFID readers can cost around \$20. These should be used to check prices, so there should be one for each Shopping Assistance Terminal, which in total should be 10. Larger RFID readers can cost around \$200 each. There should be one for each basket for items to be reshelfed, for each bagging area, and for the return terminal. This should be 17. In total, this is approximately \$3,600.

Optionally, for the convenience of employees, store tablets or smartphones may be provided to run the application (as opposed to the employee's personal device). Assuming there are 30 employees working at all times, this could cost around \$1,500.

## Section 1.2: Glossary of Terms

**Checkout Terminal:** This is a set of a tablet, an RFID scanning bagging area, and an RFID scanning basket. The bagging area will have plastic bags for customers to use and an RFID scanner to detect all the items in the bagging area. The RFID scanning basket will be a basket with an RFID scanner that can determine the items that need to be reshelfed.

**Customer Application:** This is an application that will allow customers to find items and check prices of items.

**Employee Application:** This is an application to be installed on an employee's smart device. It will synchronize tasks between employees and alert employees of tasks.

**Manager Application:** This is an application to be installed on a manager's smart device. It will allow managers to monitor employees.

**Returns Terminal:** This is a set of a tablet and an RFID scanning basket. The tablet will contain an application for returns. The RFID scanning basket will be a basket with an RFID scanner that can determine the items that need to be reshelfed.

**Shopping Assistance Terminal:** This is a set of a tablet, RFID scanner, and a RFID scanning basket to be located at the end of each aisle. The tablet will contain the Customer Application.



The RFID scanning basket will be a basket with an RFID scanner that can determine the items that need to be reshelfed.

## Section 2: System Requirements

### Section 2.1: Enumerated Functional Requirements

#### Customer Application/Customer Assistance Terminal

Identifier	Priority (5 is highest priority)	Requirement
REQ-01	4	The Customer Application will be able to check the price of a held item.
REQ-02	5	The Customer Application will be able to look up the location and price of any item in the store.
REQ-03	4	The Customer Application will be able to request an employee for assistance.
REQ-04	3	The Customer Application will be able to save a shopping list for registered users.
REQ-05	3	The Customer Application will be able to order a shopping list for pick up or delivery.
REQ-06	2	The Customer Application will be available to use on either in store Customer Assistance Terminals or on a user's smart device.
REQ-07	2	The Customer Assistance Terminal will be able to hold and inventory unwanted for reshelfing.
REQ-08	4	The Customer Assistance Terminal will notify an employee when items are waiting to be reshelfed.

#### Checkout Application/Checkout Terminal

Identifier	Priority (5 is highest priority)	Requirement
REQ-09	5	The checkout application will be able to determine all items in the bagging area.

REQ-10	3	The checkout application will be able to lock all current items in the bagging area so they can be moved to make room for more items.
REQ-11	5	The checkout application will be able to complete the purchase transaction.
REQ-12	4	The checkout application will be able to request an employee for assistance.
REQ-13	3	The checkout application will allow an employee to manually remove or add items.

#### **Return Application/Return Terminal**

<b>Identifier</b>	<b>Priority (5 is highest priority)</b>	<b>Requirement</b>
REQ-14	5	The return application will require an employee or manager to log into the application with their credentials before use.
REQ-15	5	The return application will be able to check all items in the return basket.
REQ-16	5	The return application will be able to refund the items to the customer who returned them.
REQ-17	1	The return application will check the receipt to ensure those items were bought by that customer.
REQ-18	3	The return application will be able to notify an employee when there are items to be put back on shelves after a return.

#### **Employee Application**

<b>Identifier</b>	<b>Priority (5 is highest priority)</b>	<b>Requirement</b>
REQ-19	5	The Employee Application will require all employees to log into the application with their credentials before use.
REQ-20	3	The Employee Application will log employee hours by allowing employees to clock in and clock out of work.

REQ-21	3	The Employee Application will monitor item stock on shelves.
REQ-22	5	The Employee Application will maintain a list of tasks that have been done and still need to be done.
REQ-23	3	The Employee Application will create a task and notify employees when shelves need to be restocked.
REQ-24	5	The Employee Application will create a task and notify employees when items need to be reshelfed from the Customer Assistance Terminals or the Return Terminals.
REQ-25	1	The Employee Application will create a task and notify employees when items are about to expire and need to be removed from shelves.
REQ-26	5	The Employee Application will allow employees to view tasks that still need to be done.
REQ-27	4	The Employee Application will allow employees to claim a task to let other employees know they are doing that task.
REQ-28	3	The Employee Application will allow employees to mark a task as complete.
REQ-29	2	The Employee Application will allow employees to call a manager for assistance.
REQ-30	2	The Employee Application will play a clear alert tone when a new task is available.

### Manager Application

Identifier	Priority (5 is highest priority)	Requirement
REQ-31	5	The Manager Application will require all managers to log into the application with their credentials before use.
REQ-32	3	The Manager Application will monitor stock in inventory.
REQ-33	4	The Manager Application will allow managers to view and correct employee work hours.
REQ-34	5	The Manager Application will allow managers to view the history of tasks from the employee application and their completion.

REQ-35	3	The Manager Application will alert managers if there is low stock in the inventory.
REQ-36	2	The Manager Application will alert managers when an employee needs assistance.
REQ-37	2	The Manager Application will allow managers to mark that they are going to assist an employee.
REQ-38	4	The Manager Application will calculate shop statistics such as sales and returns figures.
REQ-39	5	The Manager Application will allow managers to manually add tasks to the employee task list.
REQ-40	5	The Manager Application will calculate an employee's salary.

## Section 2.2: Enumerated Nonfunctional Requirements

### Customer Application/Customer Assistance Terminal

Identifier	Priority (5 is highest priority)	Requirement
REQ-41	4	The Customer Assistance Terminal should be clearly visible and easy to spot in the store.
REQ-42	4	The Customer Assistance Terminal should allow simple price checking by only requiring the item be held near the RF sensor.
REQ-43	4	The Customer Assistance Terminal should accept unwanted items by simply placing them into the reshelving basket.

### Checkout Application/Checkout Terminal

Identifier	Priority (5 is highest priority)	Requirement
REQ-44	4	The Checkout Terminal and application should provide clear and easy instructions on how to checkout.

### Return Application/Return Terminal

<b>Identifier</b>	<b>Priority (5 is highest priority)</b>	<b>Requirement</b>
REQ-45	4	The Return Terminal and application should clearly display items being returned for easy verification with the employee processing the return.

#### **Employee Application**

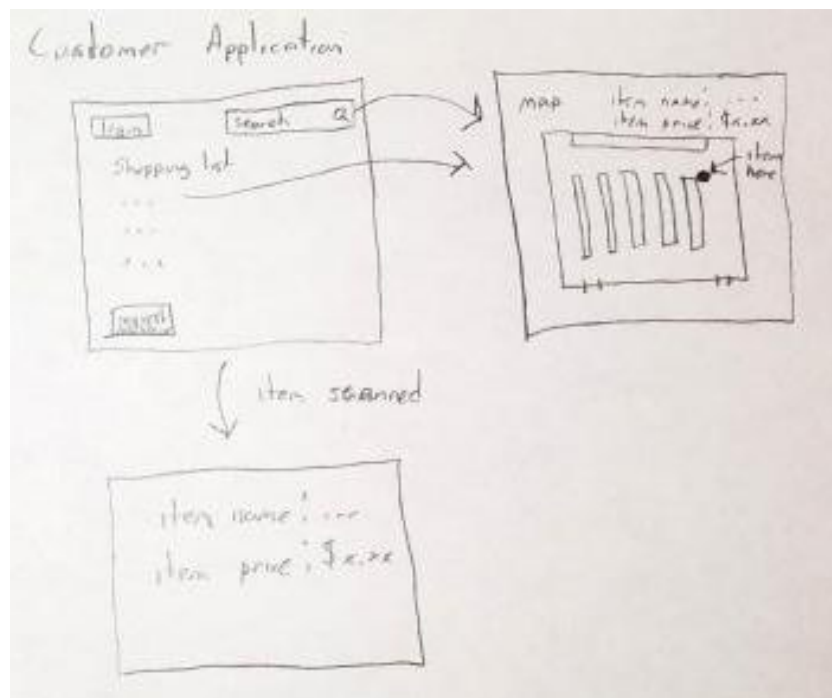
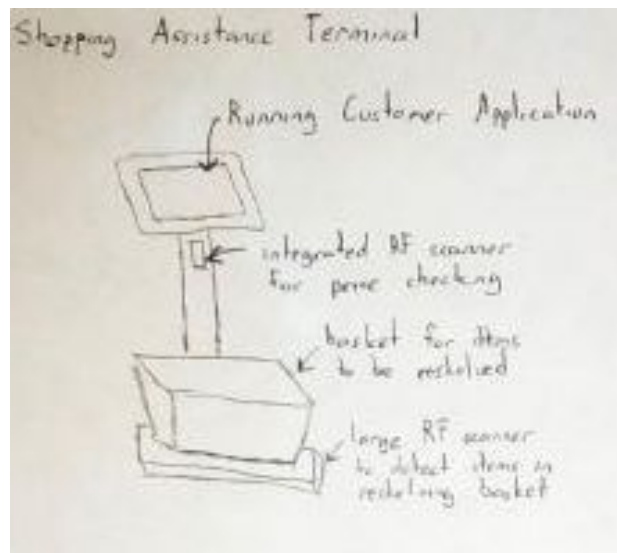
<b>Identifier</b>	<b>Priority (5 is highest priority)</b>	<b>Requirement</b>
REQ-46	5	The Employee Application should make it clear from the home screen what new tasks need to be done.
REQ-47	3	The Employee Application should be streamlined as to not busy employees with using the application.

#### **Manager Application**

<b>Identifier</b>	<b>Priority (5 is highest priority)</b>	<b>Requirement</b>
REQ-48	5	The Manager Application should be able to easily view working tasks and tasks that still need to be done to ensure tasks are being completed in a timely manner.

## Section 2.3: User Interface Requirements

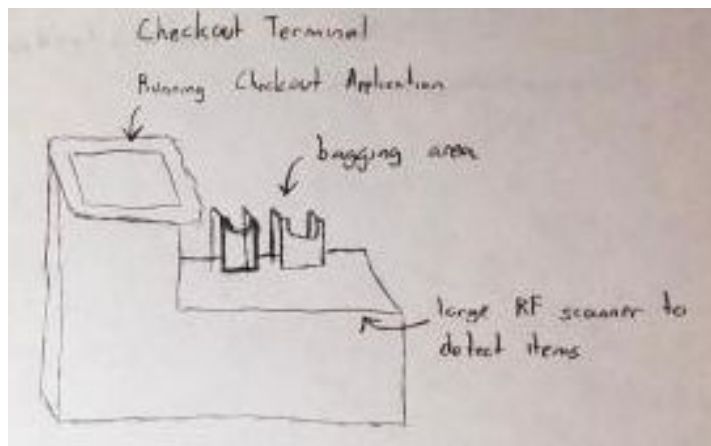
### Customer Application/Customer Assistance Terminal



Identifier	Priority (5 is highest priority)	Requirement
REQ-49	4	The Customer Assistance Terminal should have a clearly marked basket for unwanted items.

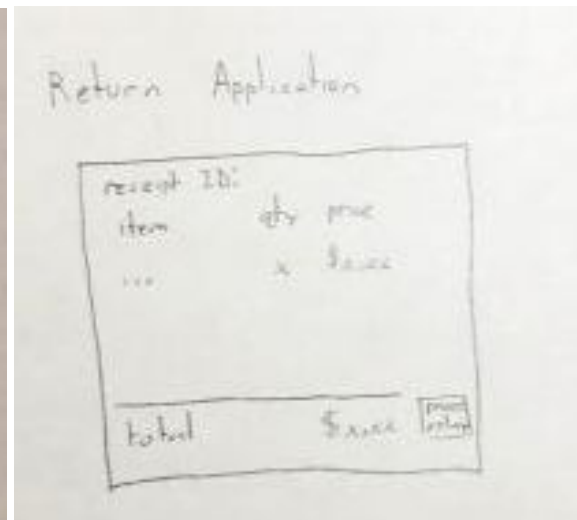
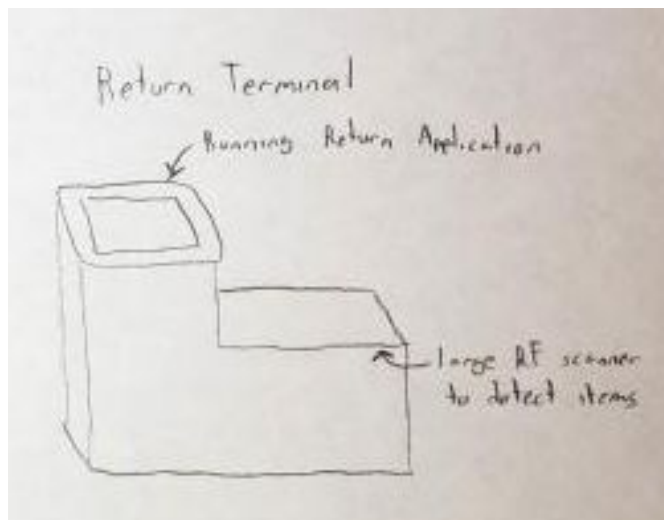
REQ-50	5	The Customer Application should have a search box for items on the home page.
REQ-51	5	The Customer Application should display the location and price of an item if it is searched for.
REQ-52	5	The Customer Application should display the price and name of an item if it is scanned for a price check.
REQ-53	4	The Customer Application should display the location and price of an item selected from a customer's shopping list.

### Checkout Application/Checkout Terminal



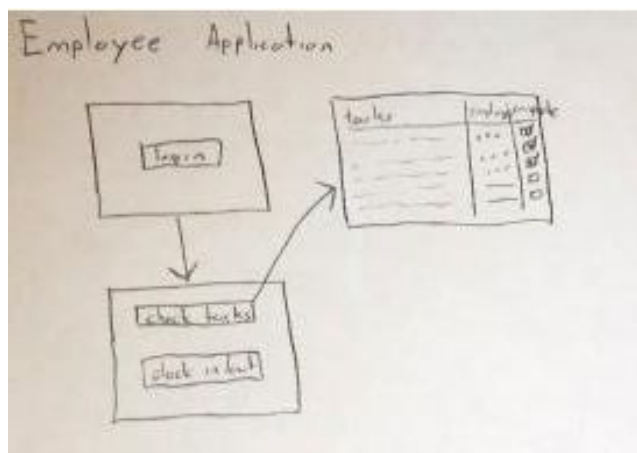
Identifier	Priority (5 is highest priority)	Requirement
REQ-54	4	The Checkout Terminal should have clear buttons for “checkout”, “lock cart” and “call for assistance”.
REQ-55	5	The Checkout Terminal should display the name, price, and amount of all products that have been scanned.
REQ-56	3	The Checkout Terminal should have a clearly marked basket for unwanted items.

## Return Application/Return Terminal



Identifier	Priority (5 is highest priority)	Requirement
REQ-57	3	The Return Terminal should display all items on a scanned receipt and allow the items to be selected.

## Employee Application

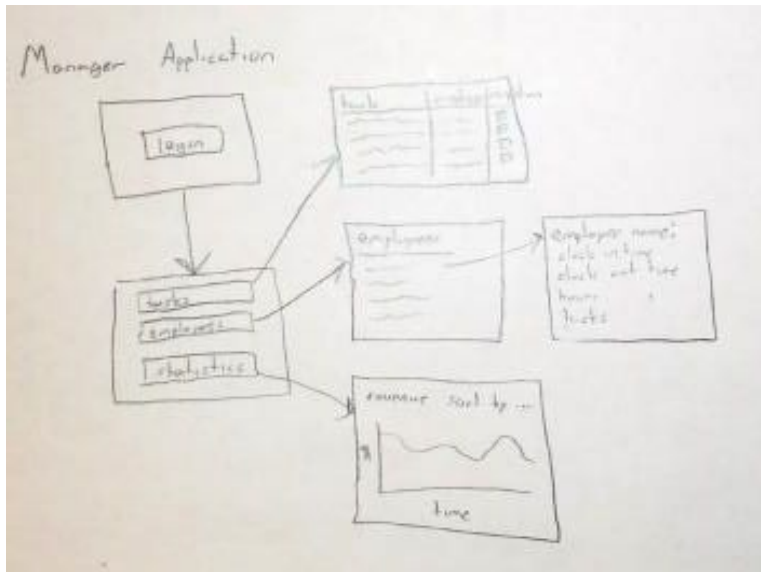


Identifier	Priority (5 is highest priority)	Requirement
REQ-58	5	The Employee Application will display all unclaimed, uncompleted tasks and will have buttons to allow an employee to claim that task.



REQ-59	3	The Employee Application will display an employee's claimed task(s) for that employee.
REQ-60	4	The Employee Application will have buttons to allow an employee to mark their claimed task(s) as complete.

## Manager Application



Identifier	Priority (5 is highest priority)	Requirement
REQ-61	5	The Manager Application will have buttons so that managers can choose to display tasks, employees, or statistics.
REQ-62	5	The Manager Application will display all tasks, their state of completion, and the employee who claimed the task (if applicable) in the task menu.
REQ-63	3	The Manager Application will display employee names in the employee menu.
REQ-64	3	The Manager Application will display an employee's clock in time, clock out time (if applicable), hours for the day (if applicable), claimed and completed tasks when an employee is selected in the employee menu.
REQ-65	3	The Manager Application will display sorted revenue statistics in the statistics menu.

## Section 3: Functional Requirements Specification

### Section 3.1: Stakeholders

The stakeholders for the store automation system are supermarket owners, customers, and employees. Supermarket owners will use this system to automate their stores and expand their business. Supermarket customers will use tablets placed in the store or use a phone application. The supermarket employees will have to learn to use the Employee Application and depending on their position, the Manager Application as well in order to best assist customers.

### Section 3.2: Actors and Goals

- **Customer Application (Initiating)** - This application will assist customers in finding an item and checking its price. Also, the application can call for an employee's help and will send them an alert for reshelving items. The application will be able to save a customer's shopping list.
- **Checkout Application (Initiating)** - This application will be able to check all the items that a customer wants to buy and will complete the transaction. It will be able to call for an employee's help. Also, it allows an employee to manually add or remove items.
- **Return Application (Initiating)** - This application will be able to process all returned items. It will check all items and check if it satisfies the return policy set by the store. After that it will give the customer a refund on the item. Also, it will notify an employee to put the returned item back.
- **Employee Application (Initiating)** - This application will monitor an employee's hours and pay. It will show what tasks employees need to do and what still remains.
- **Manager Application (Initiating)** - This application will oversee everything, especially the checking of stock. Furthermore, it will help calculate total revenue after each week and total cost to operate the store. It will monitor employee's work schedules and will be able to make any changes. It will notify the manager if an employee needs extra help.
- **Customers (Participating)** - They will buy products from the store. They will use the Customer Application to check prices and find an item that they want. After they get everything they wanted, they will use the checkout application to finish their transaction. Also, they will be able to return an item at a Return Terminal.
- **Employees (Participating)** - The employees will assist customers if they need help. They will reshelve items and put the returned items on shelves. They will know how much money they are making.
- **Manager (Participating)** - The manager will oversee everything that goes in the store. The manager will know how much inventory is in the store and will be able to order more products if any are out of stock. Also, the manager will assign employees tasks to complete.

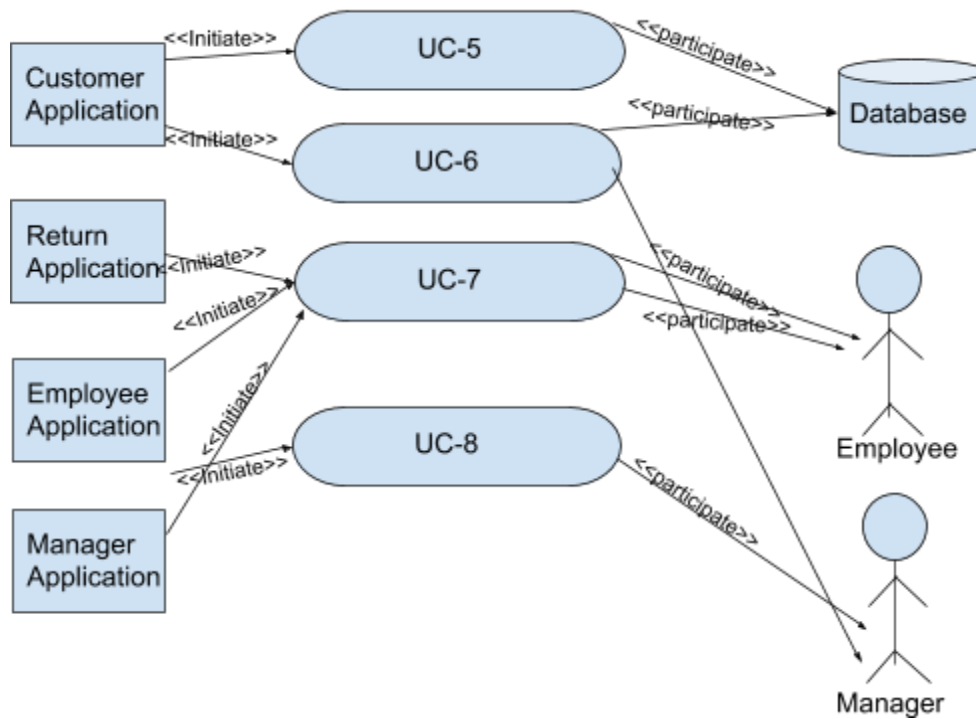
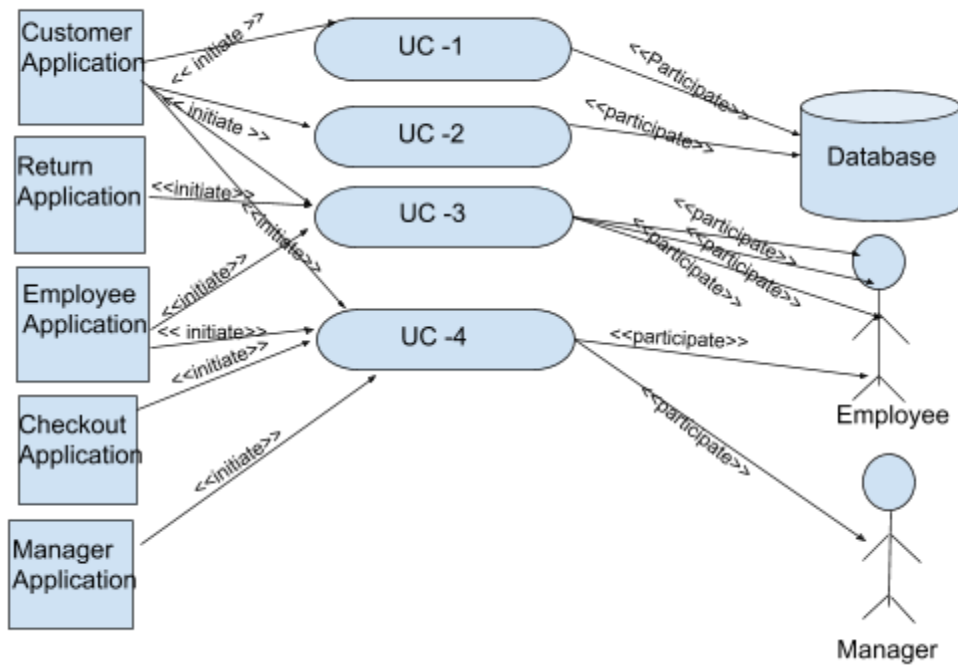
## Section 3.3: Use Cases

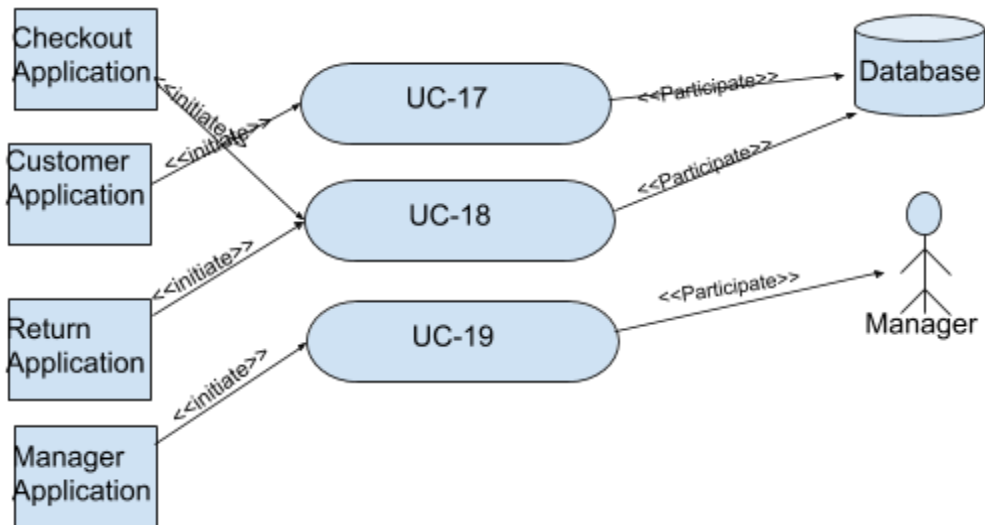
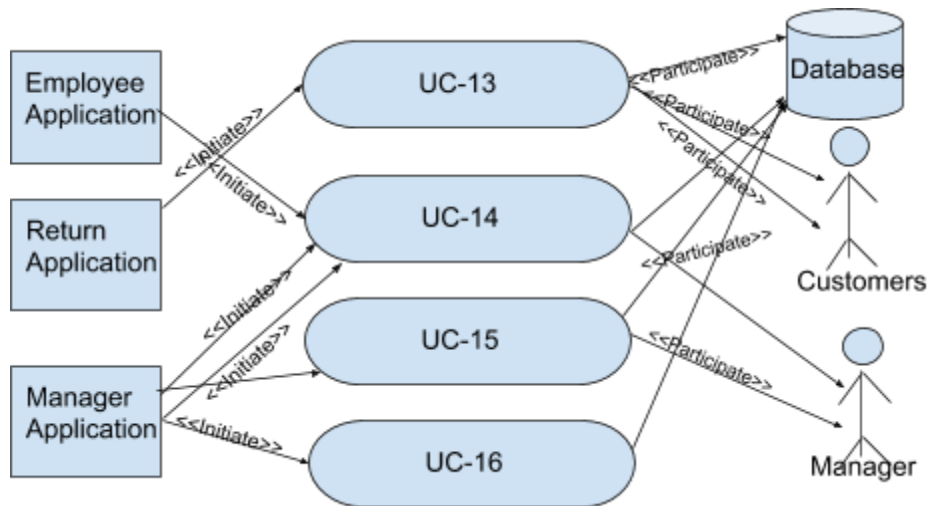
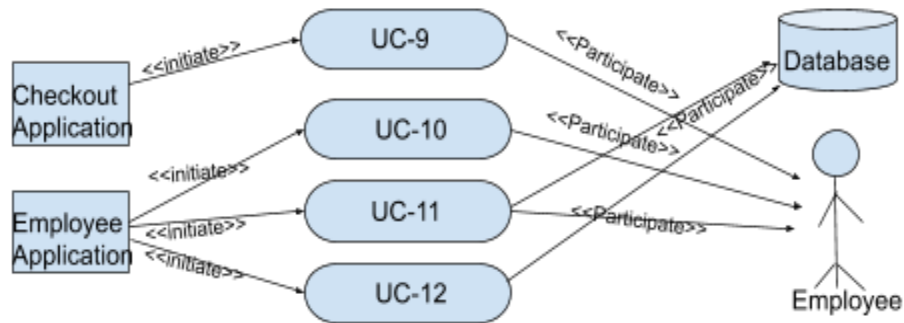
### Section 3.3.1: Casual Description

Use Case	Description	REQ
PriceChecker (UC-1)	To check and display the price of an item after scanning an item's RF tag with a sensor	REQ-1 RE-42 REQ-52
ItemLocator (UC-2)	To search for and display the location of any item from the customer assistant terminal.	REQ-2 REQ-50 REQ-51
ItemReshelve (UC-3)	To alert employees when an item needs to be reshelfed and shelves restocked	REQ-8 REQ-18 REQ-23
AssistantCaller (UC-4)	To alert employees or managers when a customer is in need of assistance	REQ-3 REQ-12 REQ-29 REQ-36
ShoppingCart (UC-5)	To store a list of items the customer wishes to purchase in a virtual shopping cart on the customer application.	REQ-4
OrderItems (UC-6)	To place an order of items for pick-up or delivery through the customer application	REQ-5 REQ-53
LogIn (UC-7)	To allow access to employee and manager applications only when an employee or manager enters their credentials	REQ-14 REQ-19 REQ-31
Checkout (UC-8)	To facilitate smooth purchase transactions through the checkout application	REQ-11 REQ-44 REQ-55
EmployeeOverride (UC-9)	To allow employees the ability to manually add or remove items from the bagging area	REQ-13
CreateTask (UC-10)	To add task to the task board of required jobs such as reshelving items or discarding expired items	REQ-24 REQ-25

TaskBoard (UC-11)	To provide a simple interface for employees to view, claim, or mark complete tasks.	REQ-22 REQ-26 REQ-27 REQ-28 REQ-58 REQ-59 REQ-60
TaskAlert (UC-12)	To send an audible alert to employees when new tasks are available and uncompleted	REQ-30 REQ-46
ReturnItem (UC-13)	To facilitate an efficient return of items process, which offers transparency of the process to the user and mechanisms to speed up multiple returns	REQ-16 REQ-17 REQ-45 REQ-57
InventoryCheck (UC-14)	To display the inventory of stock in the store and alert employees if inventory is low	REQ-21 REQ-32 REQ-35
MonitorEmployee (UC-15)	To keep track of employees' schedules and calculate salaries based on their hours	REQ-33 REQ-40
StoreStatistics (UC-16)	To calculate the total revenue and total operation cost for a given day	REQ-38 REQ-65
UnwantedItem (UC-17)	To accept items that a customer does not want until the items are reshelfed	REQ-7 REQ-43 REQ-56
ItemCheck (UC-18)	To identify items obtained by the customers.	REQ-9 REQ-15
ManagerBusy (UC-19)	To mark that the manager is helping someone in order to inform employees that the manager is busy.	REQ-37

### Section 3.3.2: Use Case Diagrams





### Section 3.3.3: Traceability Matrix

REQ-#	PW	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14	UC-15	UC-16	UC-17	UC-18	UC-19
REQ-1	4	X																		
REQ-2	5		X																	
REQ-3	4				X															
REQ-4	3					X														
REQ-5	3						X													
REQ-6	2																			
REQ-7	2																	X		
REQ-8	4			X																
REQ-9	5																		X	
REQ-10	3																			
REQ-11	5								X											
REQ-12	4				X															
REQ-13	3									X										
REQ-14	5							X												
REQ-15	5																		X	
REQ-16	5													X						
REQ-17	1													X						
REQ-18	3			X																
REQ-19	5							X												
REQ-20	3																			
REQ-21	3														X					
REQ-22	5											X								

REQ-23	3			X																
REQ-24	5									X										
REQ-25	1									X										
REQ-26	5										X									
REQ-27	4										X									
REQ-28	3										X									
REQ-29	2				X															
REQ-30	2											X								
REQ-31	5							X												
REQ-32	3													X						
REQ-33	4														X					
REQ-34	5																			
REQ-35	3													X						
REQ-36	2				X															
REQ-37	2																			X
REQ-38	4															X				
REQ-39	5																			
REQ-40	5														X					
REQ-41	4																			
REQ-42	4	X																		
REQ-43	4																X			
REQ-44	4							X												
REQ-45	4												X							
REQ-46	5											X								
REQ-47	3																			



REQ-48	5																			
REQ-49	4																			
REQ-50	5		X																	
REQ-51	5		X																	
REQ-52	5	X																		
REQ-53	4						X													
REQ-54	4																			
REQ-55	5							X												
REQ-56	3																	X		
REQ-57	3												X							
REQ-58	5										X									
REQ-59	3										X									
REQ-60	4										X									
REQ-61	5																			
REQ-62	5																			
REQ-63	3																			
REQ-64	3																			
REQ-65	3																X			
Max PW		5	5	4	4	3	4	5	5	3	5	5	5	5	3	5	4	4	5	2
Total PW		13	15	10	12	3	7	15	14	3	6	6	7	13	9	9	7	9	10	2

### Section 3.3.4: Fully-Dressed Descriptions

<b>UC-02</b>	<b>ItemLocator</b>
<b>Related Requirements</b>	2, 50, 51
<b>Initiating Actor</b>	Customer
<b>Actor's Goal</b>	To receive information about an item's location in the store
<b>Participating Actors</b>	Customer
<b>Preconditions</b>	Action performed through the customer assistance terminals
<b>Postconditions</b>	User receives map and aisle data of requested item
<b>Flow of Events (Typical Scenario Using Search Box)</b>	
→ 1.	User enters the search box
← 2.	While user searches, best-match items are displayed in a dropdown. Only items on stock are available. If an item is out of stock, the item name will appear greyed out with a symbol besides it to symbolize its status as out of stock.
→ 3.	User selects best match from dropdown menu
↻ 4.	System searches for item data
← 5.	A very simple map of the store is shown with the aisle of the item highlighted. Aisle name and number is displayed along with the map.

<b>UC-07</b>	<b>LogIn</b>
<b>Related Requirements</b>	14, 19, 31
<b>Initiating Actor</b>	Employee/Manager
<b>Actor's Goal</b>	Allow access to employee and manager applications only when an employee or manager enters their credentials
<b>Participating Actors</b>	Employee, Manager
<b>Preconditions</b>	Employee and Manager applications features are unusable
<b>Postconditions</b>	Employee and Manager applications features are usable
<b>Flow of Events (First time employee login)</b>	
→ 1.	User opens application
← 2.	Application presents login screen with suboption for users(employees) to create a new account
→ 3.	User selects “create account”
← 4.	App requests user to enter credentials for account creation
→ 5.	User inputs credentials and sends a request for account to be approved by manager
← 6.	Manager receives request from employee to create an account
→ 7.	Manager accepts request from employee
← 8.	Employee app grants employee access to employee features.
<b>Flow of Events (Login with pre-existing credentials)</b>	
→ 1.	User opens application
← 2.	Application presents login screen with an input text field
→ 3.	User inputs credentials
↻ 4.	System verifies credentials
← 5.	On verification completion, user is given access to account specific features

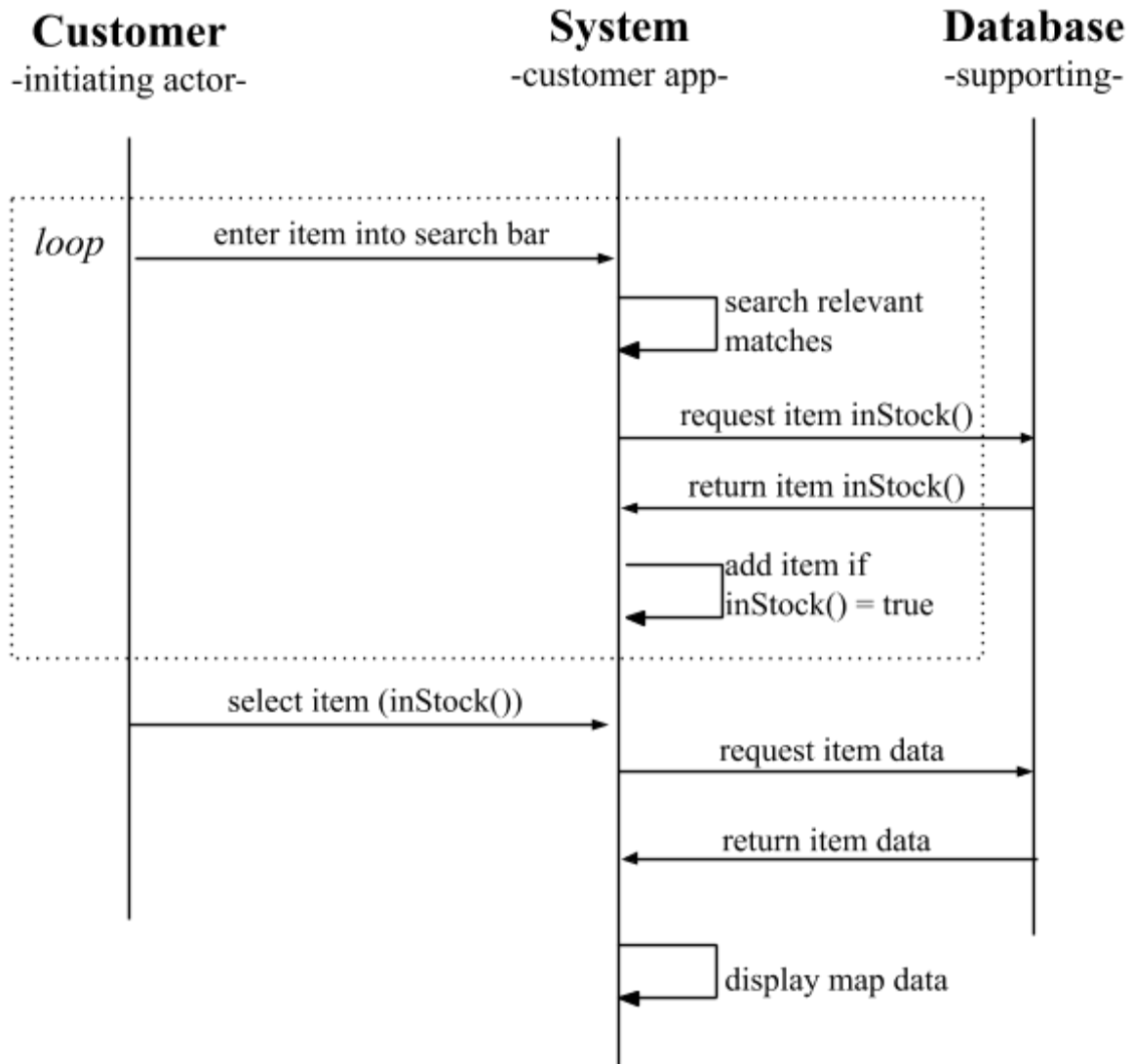
<b>UC-08</b>	<b>Checkout</b>
<b>Related Requirements</b>	11, 44, 55
<b>Initiating Actor</b>	Customer
<b>Actor's Goal</b>	To efficiently checkout items for purchase
<b>Participating Actors</b>	Employee
<b>Preconditions</b>	Customer arrives at a checkout terminal to begin checkout
<b>Postconditions</b>	Customer's items have been purchased and the necessary inventory data is updated.
<b>Flow of Events (Successful Checkout Scenario)</b>	
→ 1.	Customer presses button to begin checkout
← 2.	Terminal prompts customer to bag each item
↻ 3.	During bagging, RFID reader scans items in bagging area
← 4.	For each item in bagging area, the terminal displays the item's name, price, and quantity (if multiple of the same item is scanned), and puts the item in a list of total items for purchase.
→ 5.	Customer presses a button indicating the checkout is finished
← 6.	Terminal prompts user to select payment method (cash or credit)
↻ 7.	Once payment is processed, inventory data is updated and checkout is complete
<b>Flow of Events (Alternate Scenarios)</b>	
→ 1.	Customer presses a button to get assistance from an employee
↻ 2.	System sends task alert to alert an employee that a customer at terminal A needs assistance

<b>UC-14</b>	<b>ReturnItem</b>
<b>Related Requirements</b>	16, 17, 45, 57
<b>Initiating Actor</b>	Customer
<b>Actor's Goal</b>	To return purchased items with a refund
<b>Participating Actors</b>	Employee, Customer, Return Terminal (System)
<b>Preconditions</b>	Customer must provide proof of purchase through receipt
<b>Postconditions</b>	Item having met return policy is returned to shelves
<b>Flow of Events (Successful Return Scenario)</b>	
→ 1.	Customer hands receipt and item to employee
→ 2.	Employee scans receipt
← 3.	System lists items purchased by customer as recorded on receipt
→ 4.	Employee scans items
↻ 5.	System matches items to the receipt list, confirming that the customer had bought the items.
← 6.	System displays items return policy for the employee to check before approving the return
↻ 7.	On approve, items are placed in return bin and a task is created for them to be restocked
→ 8.	Employee selects whether return is done with cash or credit
← 9a.	System refunds customer and deducts amount from earnings (employee hands cash return to user)
← 9b.	System refunds customer and deducts amount from earnings once credit card transaction completed
<b>Flow of Events (Unsuccessful Return Extension)</b>	
← 5a.	If item scanned does not match one on receipt, an error is displayed and the return process is over

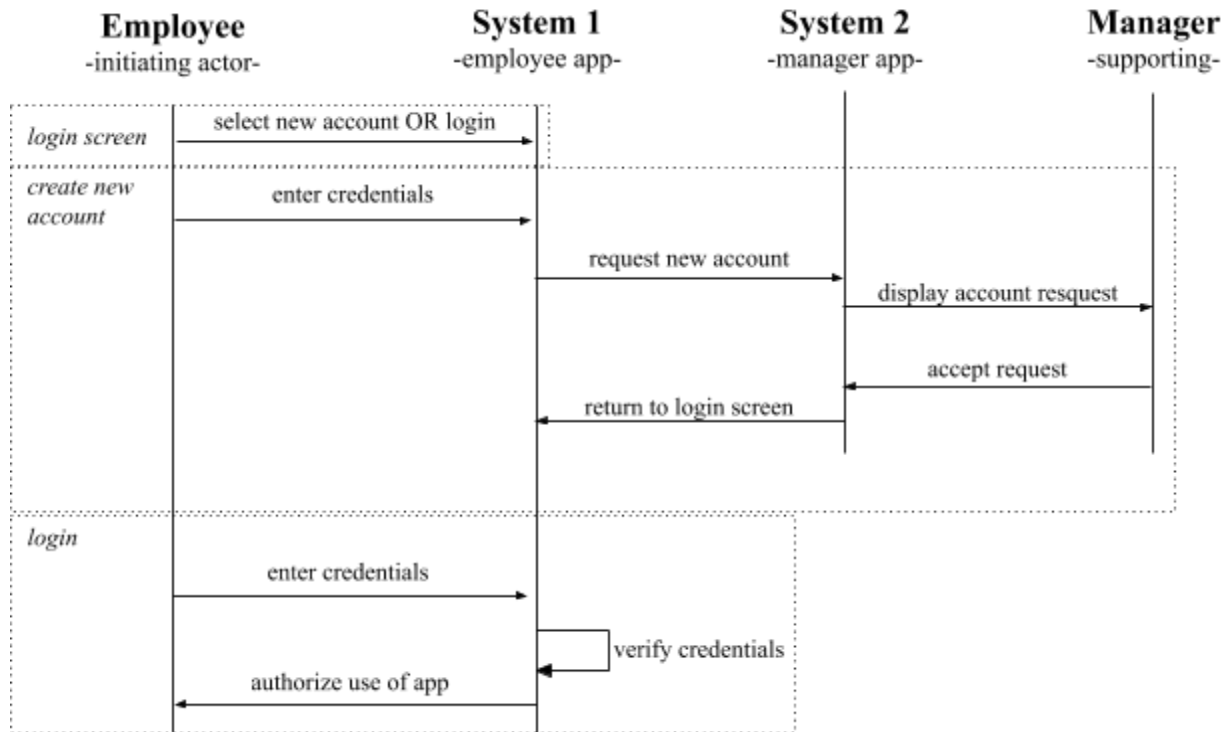
<b>UC-01</b>	<b>PriceChecker</b>
<b>Related Requirements</b>	1, 42, 52
<b>Initiating Actor</b>	Customer
<b>Actor's Goal</b>	To receive information about an item's price
<b>Participating Actors</b>	
<b>Preconditions</b>	Action performed through a customer assistance terminal
<b>Postconditions</b>	User has obtained the price of an item
<b>Flow of Events (Success Scenario)</b>	
→ 1.	User scans item in RFID scanner
↷ 2.	System searches database for item
← 3.	System returns the price of item

## Section 3.4: System Sequence Diagrams

### Use Case: ItemLocator

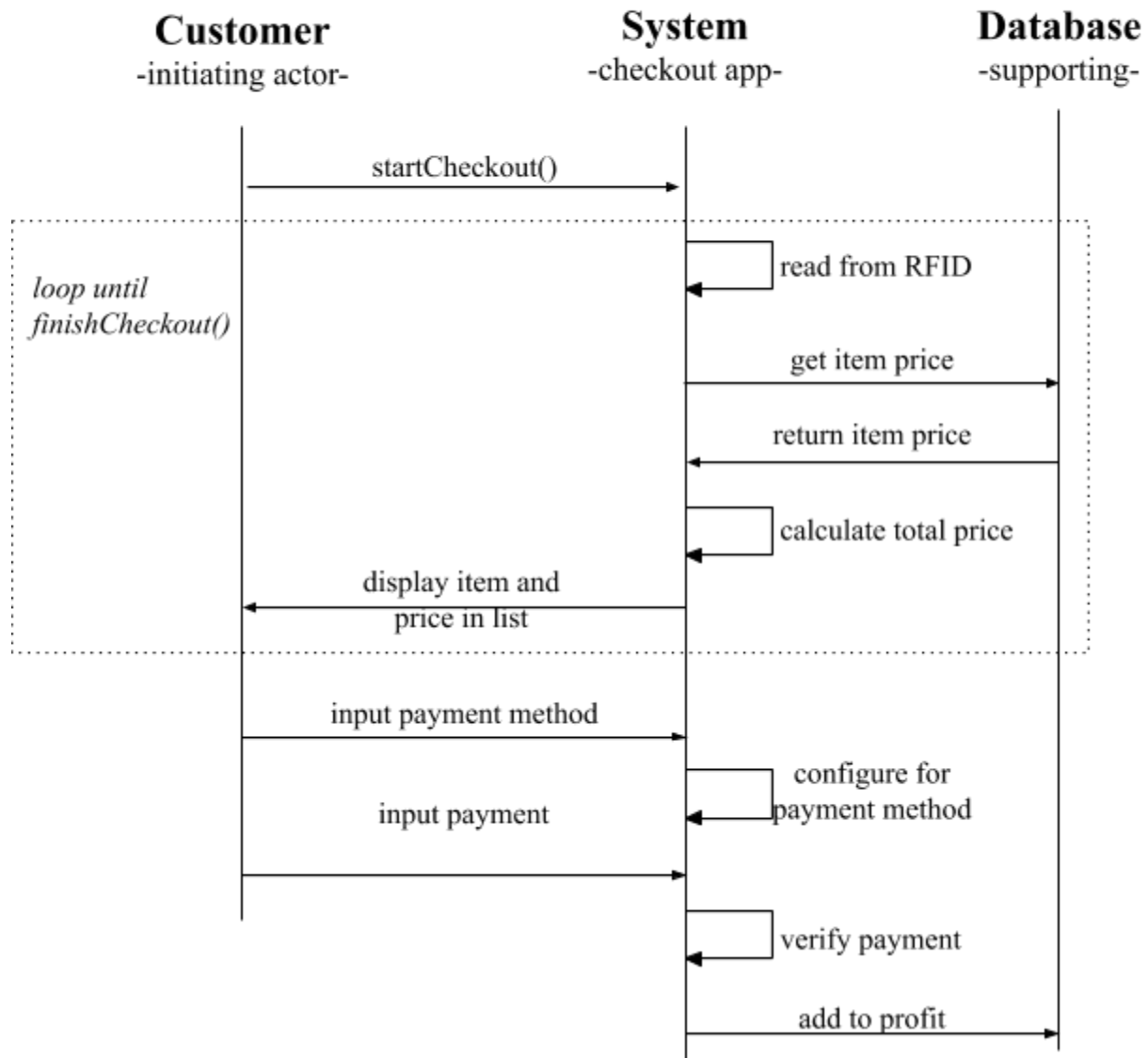


## Use Case: **LogIn**

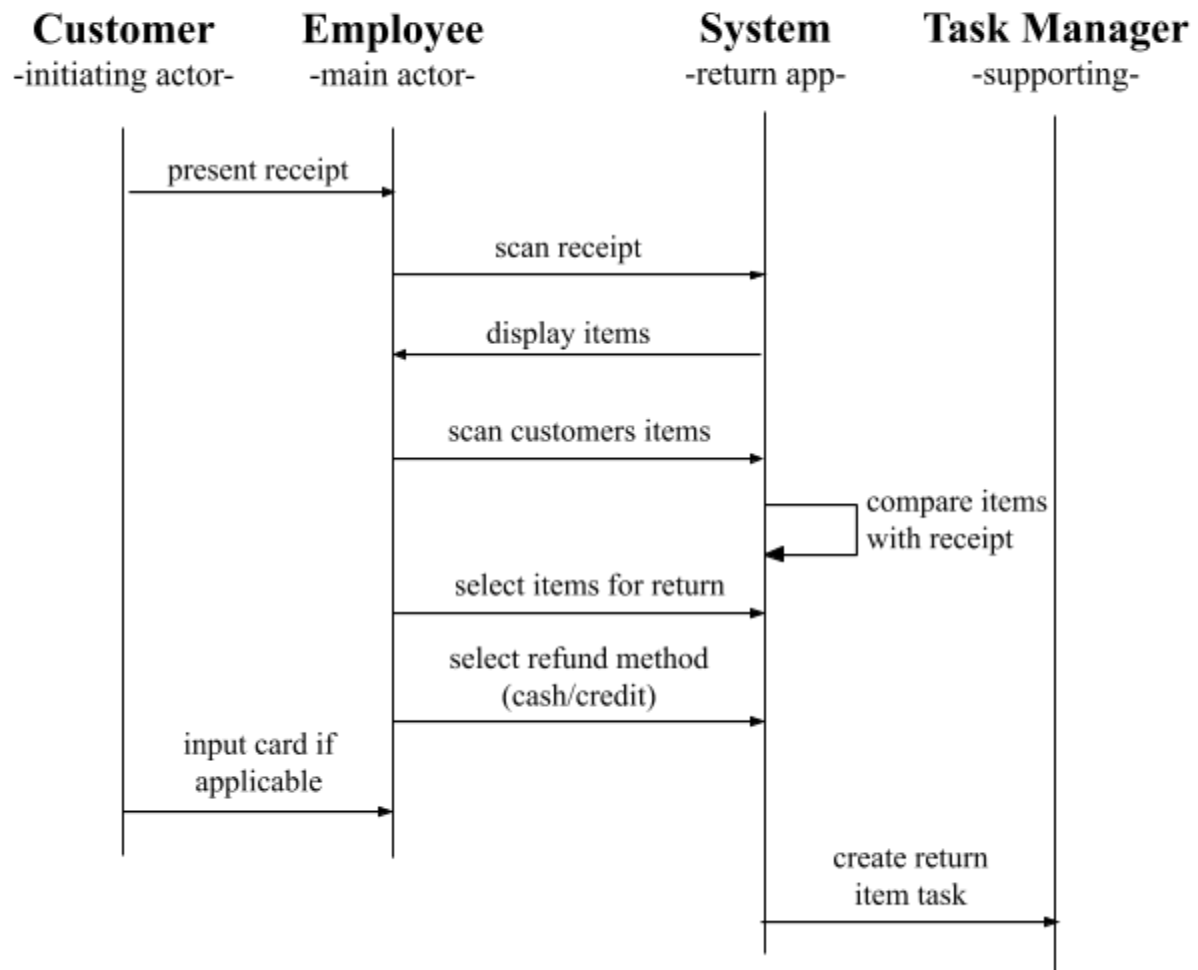




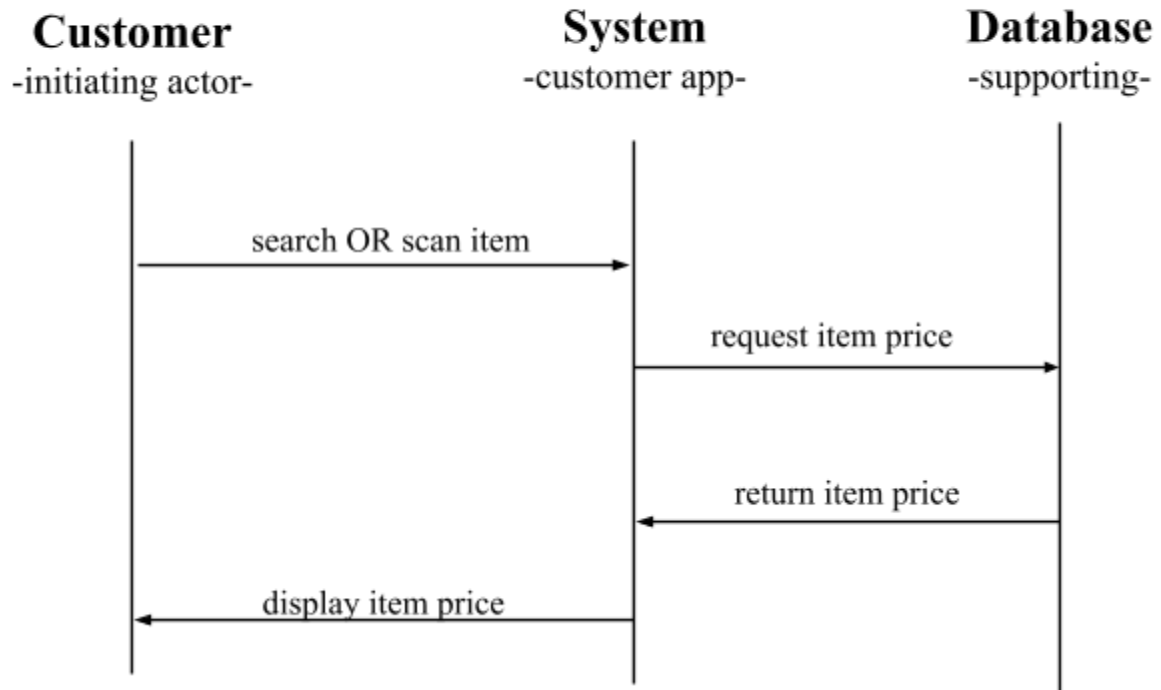
## Use Case: Checkout



## Use Case: ReturnItem



## Use Case: PriceChecker



## Section 4: User Interface Specification

### Section 4.1: Preliminary Design

#### Customer Assistance Terminal

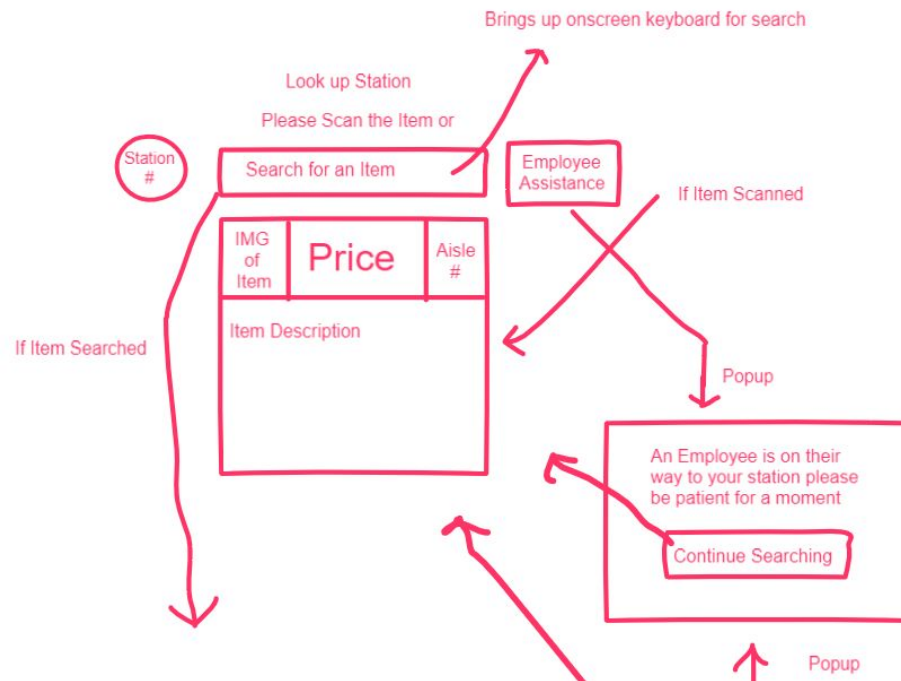


Figure 4.1-1

For the lookup station at a Customer Assistance Terminal, there are multiple paths. The first path is to scan an item in order to check the price as shown in [Figure 4.1-1](#). This is a quick RFID scan and then the app displays the relevant information. If assistance is needed then the user would press the needs assistance button. This will send an alert to an employee and that employee who gets the task up will receive a priority task added to their task list.

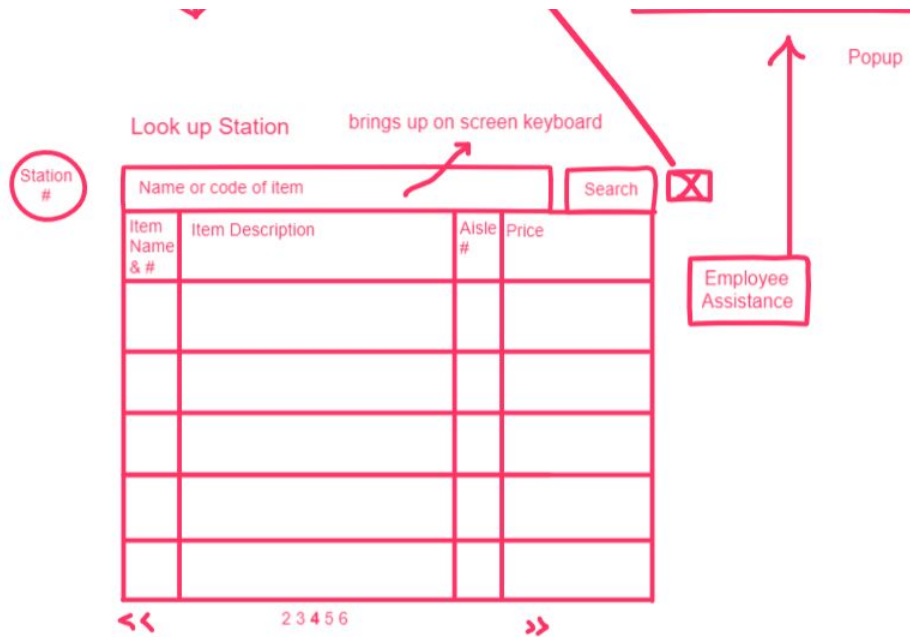


Figure 4.1-2

The other path is to use the search function in order to find an item's location in the store or to compare similar items as shown in [Figure 4.1-2](#). This is done by clicking the search bar and then searching through keyword or item number and then hitting enter. The program will then show you the search results. There is also an employee assistance button on this page that will act the same as the other button.

### Employee Application

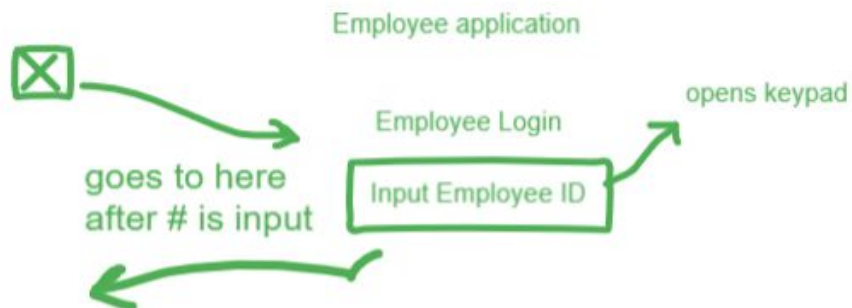


Figure 4.1-3

For the Employee application the first step is to input their employee ID. The display is shown in [Figure 4.1-3](#). This will then allow them to access 4 separate functions: clock in/clock out, check hours, check schedule, and task list. These are shown below in [Figure 4.1-4](#).

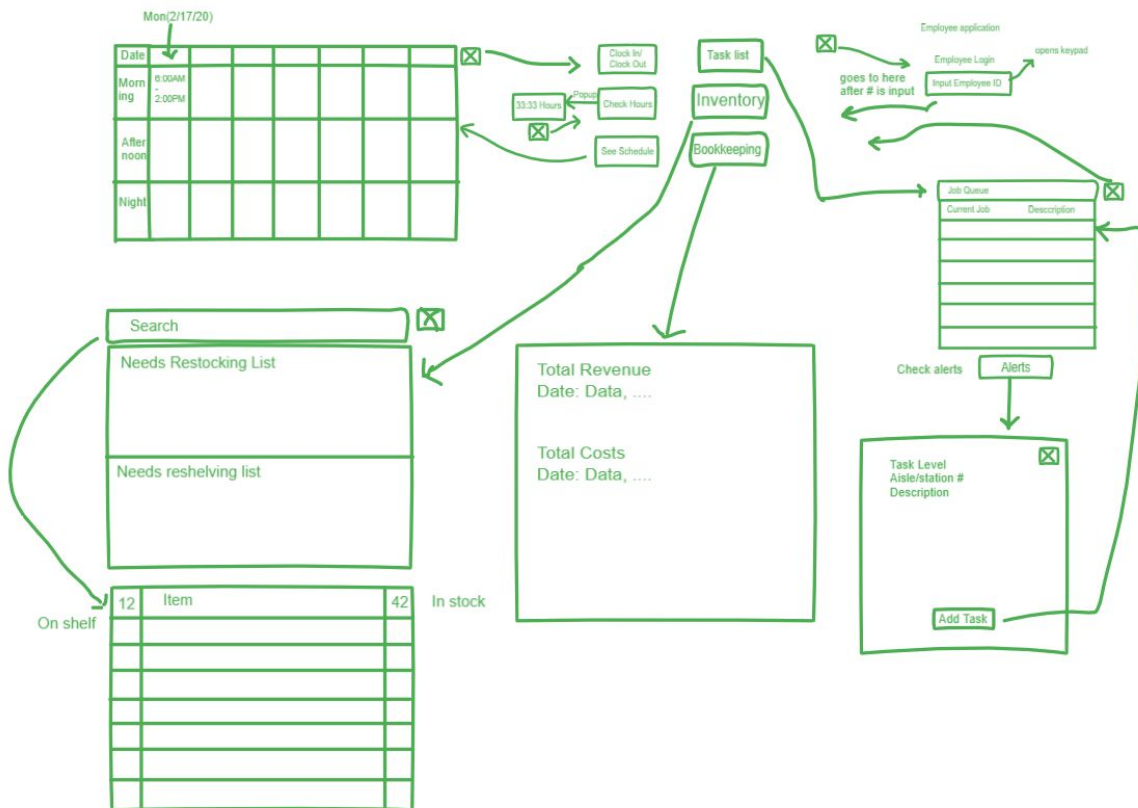


Figure 4.1-4

The clock in/clock out button will track the employees hours in the timesheet database. The check hours buttons will activate a pop up with the amount of hours worked that week, but for a manager, it will show the total hours worked for every employee. The check schedule button will show when an employee is working that week and for the manager will show who is working in any given time slot. Then on the task list screen an employee can see what their current tasks are and the priority levels of these tasks. It also has the alert button so that if they are looking for a task or a priority task shows up it can get allocated properly.

Any manager has some manager specific apps: inventory and bookkeeping. The inventory button allows a manager to see the inventory of the store and allow the allocation of tasks. First the manager would click the button and then they would see what needs to be restocked and what needs to be reshelved. Then they would be able to search for the specific stock of an item through the search bar. Finally, the bookkeeping button allows a manager to see the total revenue and costs of the store.

## Customer Application

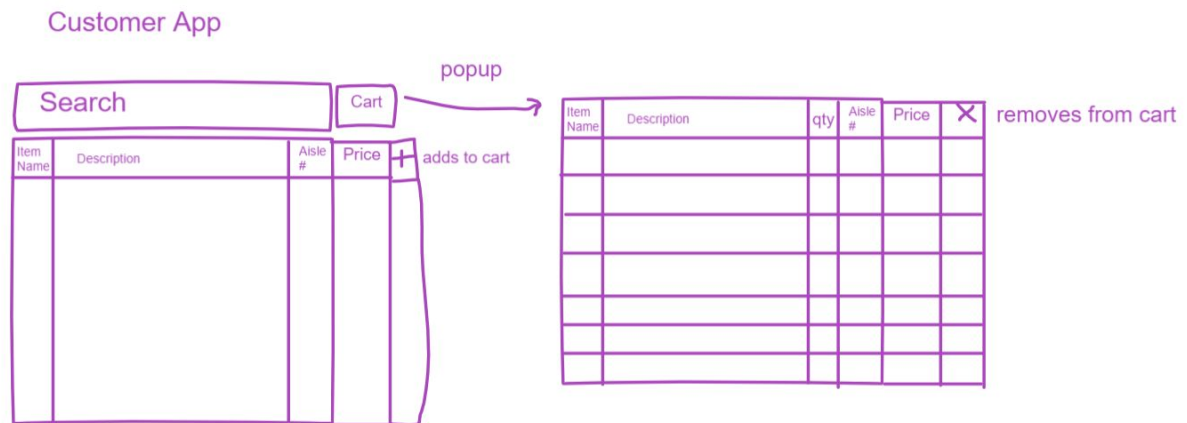


Figure 4.1-5

For the customer application we have a simple app that has the same search capability of the in store tablets but for a specific customer as seen in [Figure 4.1-5](#). It allows the user to search items and add them to their cart and remove them as necessary. It also gives the ability to see the aisle where the item is located and check the price of said item.

## Checkout Terminal



Figure 4.1-6

As stated in the project problem statement, the checkout system should be mostly employee involvement free, unlike current self-checkout stations at supermarkets. The user is initially greeted with a direction to place their bag in the RFID scanning area; this is shown in [Figure 4.1-6](#). This area will scan the RFID of all the products in a user's bag and then will automatically go to the Cart Screen.

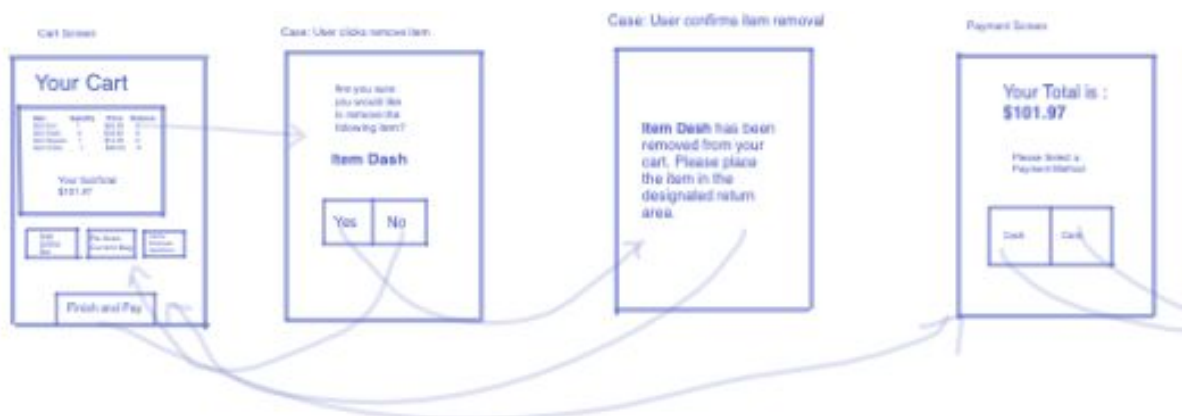


Figure 4.1-7

This screen displays all of the items that the scanner has recognized and displays the name, quantity, and price of the item along with the option to remove any specific item; the screen



should look like the leftmost screen in [Figure 4.1-7](#). The subtotal of all the products scanned is also indicated to the user. Once they see this, they have options to either scan another bag, re-scan the current bag if there are any errors present in the RFID scanning, call an employee for assistance for more specific and unexpected problems, or finish and pay.

If the user selects to scan another bag, they will be directed back to the welcome screen and their new bag's items will be added to the current cart total. If they wish to rescan the current bag, all items in the cart will be deleted and they will be led back to the welcome screen. Calling an employee for assistance will notify an employee of what system is requiring their assistance. If they select finish and pay, they will be directed to the payment screen.

It was stated earlier that the user was given options to remove specific items from their cart; this topic shall be further expanded on. If the user elects to remove an item, they are taken to a page which asks them to confirm removal of the item; if they elect to then not remove the item, the item is not removed from the cart and they will go back to the cart screen. The associated screens are the second and third screens in [Figure 4.1-7](#).

If a user elects to remove an item from the cart, then the item is deleted from the cart, and they are led to another page which directs them to place the removed item into a designated area as specified earlier in this report, and afterwards they are taken back to the cart screen.

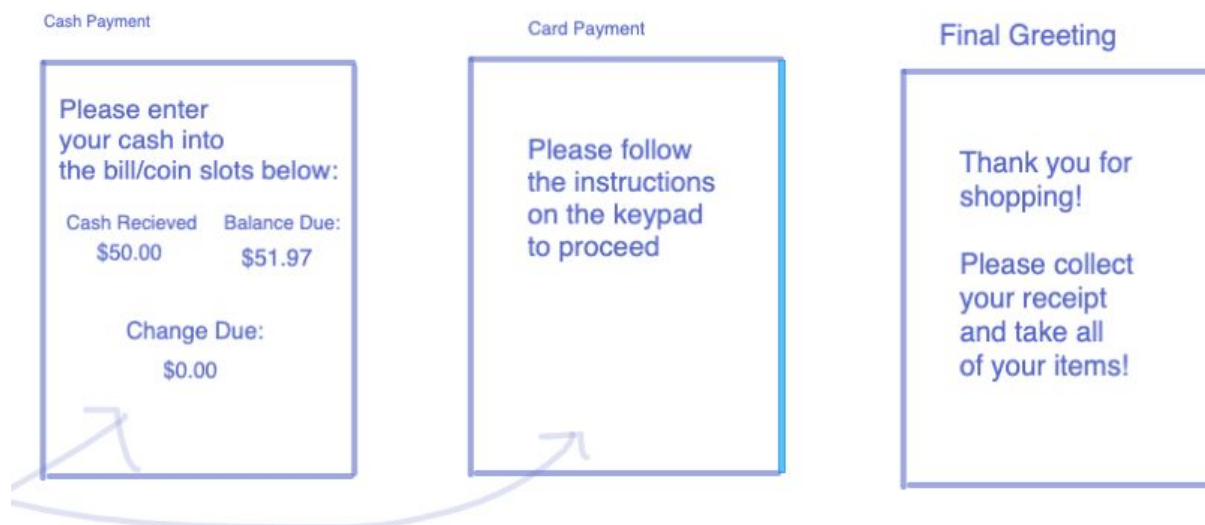


Figure 4.1-8

The case when the user selects the finish and pay button shall now be examined. When the user clicks this, then their subtotal is calculated with taxes and displayed on a screen as shown in the rightmost screen in [Figure 4.1-7](#). On the same screen, they are asked whether they would like to make a card payment or a cash payment.

If they select cash payment, they are taken to a cash payment screen and directed to enter bills/coins into the register's designated slots. This cash payment screen is shown in the leftmost screen in [Figure 4.1-8](#). The system will tell them the amount of money that has been processed, the balance that is due, and if they have paid more than their total, the change that is due to them before it takes them to the final screen.

If the user selects card payment, then they are directed to follow the instructions on the given keypad as they would for all card payments. This will be shown like on the middle screen in [Figure 4.1-8](#). After the user is done scanning and paying for all of their items, they will see the final greeting screen that thanks them for shopping and remind them to collect their receipt and all of their items.

## Returns Terminal

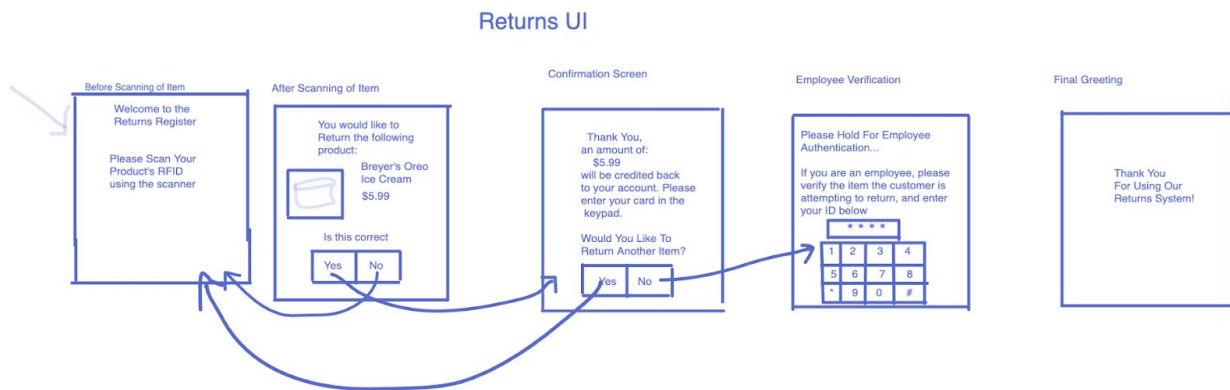


Figure 4.1-9

The process of the return terminal shall be explained. The screens involved in this case are shown in [Figure 4.1-9](#). The user will first be prompted to scan the item they wish to return using the RFID scanner. The scanner will scan the item and then display the item's name, weight, price, and in some cases, an image of the item. The user is then asked to verify whether the scanner has scanned the right item, and if not they will be asked to rescan it. If the product is scanned correctly then the user will be asked whether they would like to scan another item. If they select yes, the process restarts, and the item is added to a list, if not, then they are asked to hold for employee authentication as mentioned earlier in this report.

Although having an independent self-checkout system would be greatly beneficial, preventing cases of possible user abuse of the system is more important as it could end up costing the company a loss. The employee verification is put in place so that users can't return an empty box or scan a proxy RFID. The employee verification page asks for an employee to enter their ID, which like passwords, is hidden from the customer, and once the employee ID is confirmed, then the customer is prompted to a confirmation screen saying that the amount will be credited back to the account for which they enter their card in for, which then automatically changes to the final greeting screen.

## **Section 4.2: User Effort Estimation**

**Select several typical usage scenarios and, as you walk through the flow of events, count and report the number of mouse clicks and/or keystrokes that are needed to accomplish the task. What fraction of these goes to user-interface navigation vs. clerical data entry?**

**User wants to find what aisle cereal is in:** The user will find a lookup station and click on the search bar to bring up the keyboard. Then they will input the type of cereal and hit search. The program will show search results along with a short description, price, and aisle for each. 2 clicks, 1 keystroke.

**User needs assistance after price lookup:** User scans the item at lookup station(1 action). Then the User pushes the employee assistance button(1 click). An employee then gets a priority task notification on their app that a user needs assistance at station 00. 2 actions, 0 keystroke

**Employee wants to check their current task:** The employee opens their app and inputs their ID #(1 keystroke). Then they would click the task list button(1 click). 1 clicks, 1 keystroke

**Employee wants to check current weekly hours:** The employee opens their app and inputs their ID #(1 keystroke). Then they would click the check hours button(1 click). 1 clicks, 1 keystroke

**Manager wants to check current stock of an item:** The manager will log in to the app with his employee ID number(1 keystroke) and then he will choose the inventory button(1 click). He then uses the search bar to find the stock on the shelf and in the storeroom of said item(1 keystroke). 1 click 2 keystrokes

**User wants to check out 2 bags of items:** User places first bag in scanning area, clicks scan another bag (1 click), places second item in scanning area, clicks 'finish and pay' (1 click). User

pays with a card so that it is selected (1 click), and the user proceeds to use the keypad to finish the transaction. (overall 3 clicks)

**User wants to check out 2 bags of items, but wants to remove 1 item:** User places first bag in scanning area, sees item for removal in the cart screen and clicks the 'X' for removal (1 click). They then click to confirm removal (1 click) and then select scan another bag for their second bag (1 click) then finish and pay (1 click) then pay with cash (1 click) and finish the transaction accordingly. (overall 5 clicks)

**User wants to return 2 items:** User scans first item and verifies that it is correct (1 click), then and user says they would like to return another item (1 click), verifies it is correct (1 click), then says they would not like to scan another item (1 click) and employees comes in to enter their information (1 keystroke). (overall 4 clicks, 1 keystroke)

## Section 5: Domain Analysis

### Section 5.1: Domain Model

#### Section 5.1.1: Concept Definitions

Responsibility Description	Type	Concept
Coordinate actions of all concepts associated and interacting with a system (app/terminal)	D	Controller (customer, checkout, return, employee, manager)
Holds all information about an item	K	Item
Stores price info about an item (Database)	D	PriceHolder
Stores info about amount of each item on shelves and in storage (Database)	D	StockManager
Stores info about item's aisle number, map data (Database)	D	ItemLocator
Creates and delegates tasks (including alerts) for employee and manager	D	TaskManager
A job that needs to be completed by an employee or manager	K	Task
Holds list of all Employee objects	D	EmployeeDatabase
Database of the company's financial data (profit, transactions, etc.)	D	FinancialDatabase
Contains a list of items placed in return bins to be restocked	D	Bin
Contains a map of the store and functions that highlight different areas given an aisle and section number	D	MapMaker
Sends information to user through a user interface	D	Terminal

## Section 5.1.2: Association Definitions

Concept Pair	Association Description	Association
Controller PriceHolder	Controller requests the price of an item that was searched for or scanned	requests price
Controller (manager) StockManager	Controller requests storage information about an item that was searched for.	requests stock
Controller (checkout/return) StockManager	Controller updates storage information about an item that was purchased or returned.	updates stock
Controller ItemLocator	Controller requests location information about an item that was searched for or scanned. This info includes aisle number and map data for that item.	requests location
Controller MapMaker	Controller requests a map image with the aisle and section number highlighted for a specific item.	requests mapimage
Controller ( <i>manager</i> ) EmployeeDatabase	Controller requests employee hours	requests hours
Controller ( <i>employee</i> ) EmployeeDatabase	Controller (employee app) sends employee hours to database	update hours
Controller ( <i>manager</i> ) FinancialDatabase	Controller requests financial data	requests finances
Controller ( <i>checkout/return</i> ) FinancialDatabase	Controller sends profit	send profit
Controller TaskManager	Controller requests employee task data	views tasks
Controller TaskManager	Controller sends Task to task manager	creates task
Bin TaskManager	Bin sends Task to task manager	creates task

TaskManager Controller	Task Manager sends Task to controller (employee system)	displays tasks
Controller Terminal	Controller creates and sends to user a task alert of a new task upon receiving new task	sends alert
Controller Terminal	Controller sends info to terminal to display to user	displays UI

### Section 5.1.3: Attribute Definitions

Concept	Attributes	Attribute Description
Item	name	String name of item
	price	Price of an Item
	aisle	The aisle an Item is shelved in
	section	Position of Item in an aisle
	numInStock	Number of Items on shelves
	numInStorage	Number of Items in backroom
	numInBin	Number of Items not on shelves due to being returned or put in Bin
	binNumber	An identifier of which bin an Item has been placed in
	RFID	Unique identifier for Item
PriceHolder	Item list	List of items whose prices can be accessed
StockManager	Item list	List of items whose stock information can be accessed
ItemLocator	Item list	List of items whose location data can be accessed
Controller	login	Check if user is logged in to system
	hours	Employee system counts hours for each employee

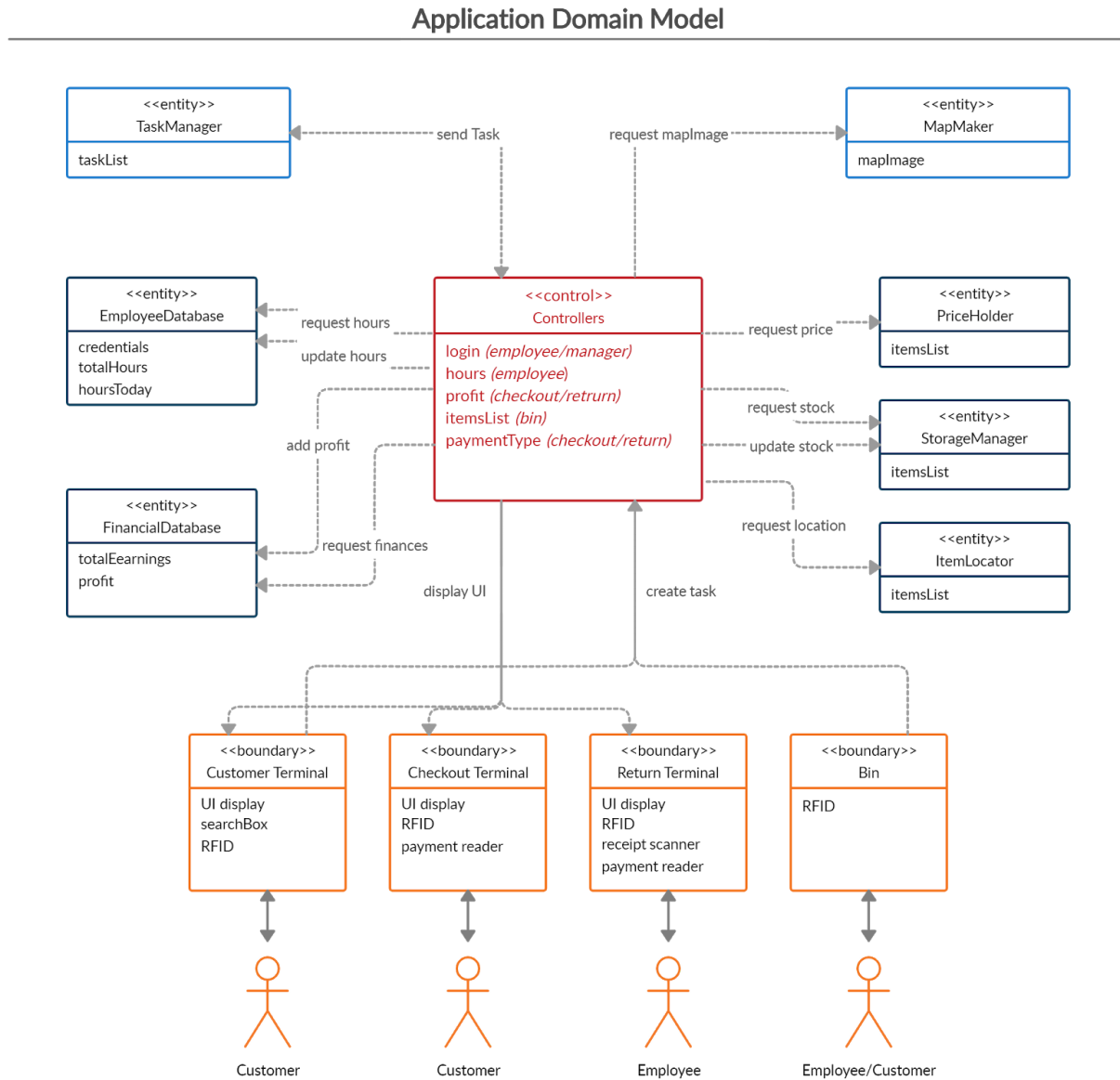
	profit	Checkout and Return systems sends profit (or loss) to financial database
Bin	Item list	List of Items in all physical bins
TaskManager	TaskList	A list of all Tasks received to display
Task	details	String explaining task
	complete	Boolean to check whether or not task is complete
	employee	Identifier of which employee is working on task
EmployeeData base	credentials	Data needed for an employee to sign in to account
	totalHours	Total hours worked by an employee
	hoursToday	Total hours worked today
FinancialData base	total earnings	total earnings of company
	today's profit	Profit for a single day
MapMaker	mapimage	An image of the store generated using location data from an Item
SearchRequest	name	name of an item



### Section 5.1.4: Traceability Matrix

		DOMAIN CONCEPTS							
UC-#	PW	Price Holder	Stock Manager	Item Locator	Map Maker	Employee Database	Financial Database	TaskManager	Terminal
UC-1	13	X	X	X			X		X
UC-2	15				X				
UC-3	10		X	X		X			
UC-4	12	X	X				X	X	
UC-5	3	X	X				X		X
UC-6	7	X	X				X		X
UC-7	15					X			
UC-8	14	X	X				X		X
UC-9	3	X	X			X	X		X
UC-10	6		X	X		X		X	X
UC-11	6					X		X	X
UC-12	7					X		X	
UC-13	13	X							
UC-14	9		X	X		X		X	
UC-15	9					X	X	X	
UC-16	7	X	X				X		
UC-17	9	X	X				X	X	
UC-18	10		X	X			X		
UC-19	2					X		X	

## Section 5.1.5: Application Domain Model



## Section 5.2: System Operation Contracts

Operation	Locate Item
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- Item name searched in searchBox</li><li>- Item stock &gt; 0</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- Item's location data is used by MapMaker to generate map image</li></ul>

Operation	Log In
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- User is currently not logged into system</li><li>- User has already created an account</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- User is granted access to role specific application</li></ul>

Operation	Checkout
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- A checkout is not currently in progress</li><li>- Customer's items have valid RFID tag</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- Customer's items are purchased (transaction successful)</li><li>- Inventory databases updated</li><li>- Unwanted items placed in return bins</li></ul>

Operation	Return Item
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- Customer possess proof of purchase (receipt)</li><li>- Item is compliant with return policy</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- Items, in accordance with return policy, are placed in return bins</li><li>- Customer receives necessary refund</li></ul>

Operation	Check Price
<b>Preconditions</b>	<ul style="list-style-type: none"><li>- Item scanned has valid RFID tag</li><li>- Item is in database</li></ul>
<b>Postconditions</b>	<ul style="list-style-type: none"><li>- User obtains the price of an item</li></ul>

## Section 6: Project Size Estimation Based on Use Case Points

	UUCW	UAW
UC-1	10	1
UC-2	10	3
UC-3	10	2
UC-4	5	3
UC-5	10	3
UC-6	5	3
UC-7	10	3
UC-8	15	3
UC-9	10	3
UC-10	5	2
UC-11	10	3
UC-12	5	2
UC-13	10	3
UC-14	10	2
UC-15	5	1
UC-16	10	1
UC-17	5	1
UC-18	15	2
UC-19	10	1
Total	170	42
UCP	226.84	

		Weights	
T1	5	2	10
T2	5	1	5
T3	5	1	5
T4	2	1	2
T5	4	1	4
T6	1	0.5	0.5
T7	5	0.5	2.5
T8	2	2	4
T9	3	1	3
T10	4	1	4
T11	5	1	5
T12	1	1	1
T13	1	1	1
TCF	1.07		

## Section 7: Plan of Work

The Gantt charts are color coded by team responsibility. The light blue color is undecided, dark blue is group A, green is group B, red is group C, and purple is groups B and C. [Figure 7-1](#), [Figure 7-2](#), and [Figure 7-3](#) are the breakdowns of the prospective responsibilities of the reports. These are based on the assumed content of the reports and not a balanced breakdown of work. The duties of each group will be revised as the project progresses and a more accurate estimate of work will be updated.

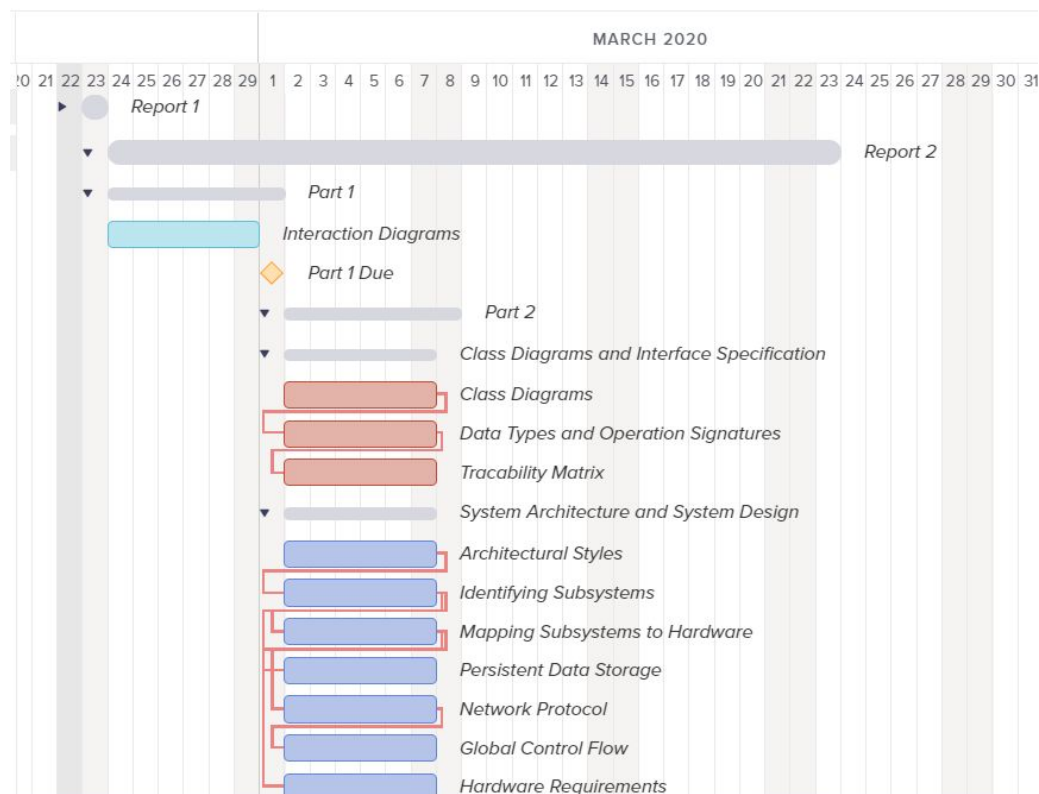


Figure 7-1

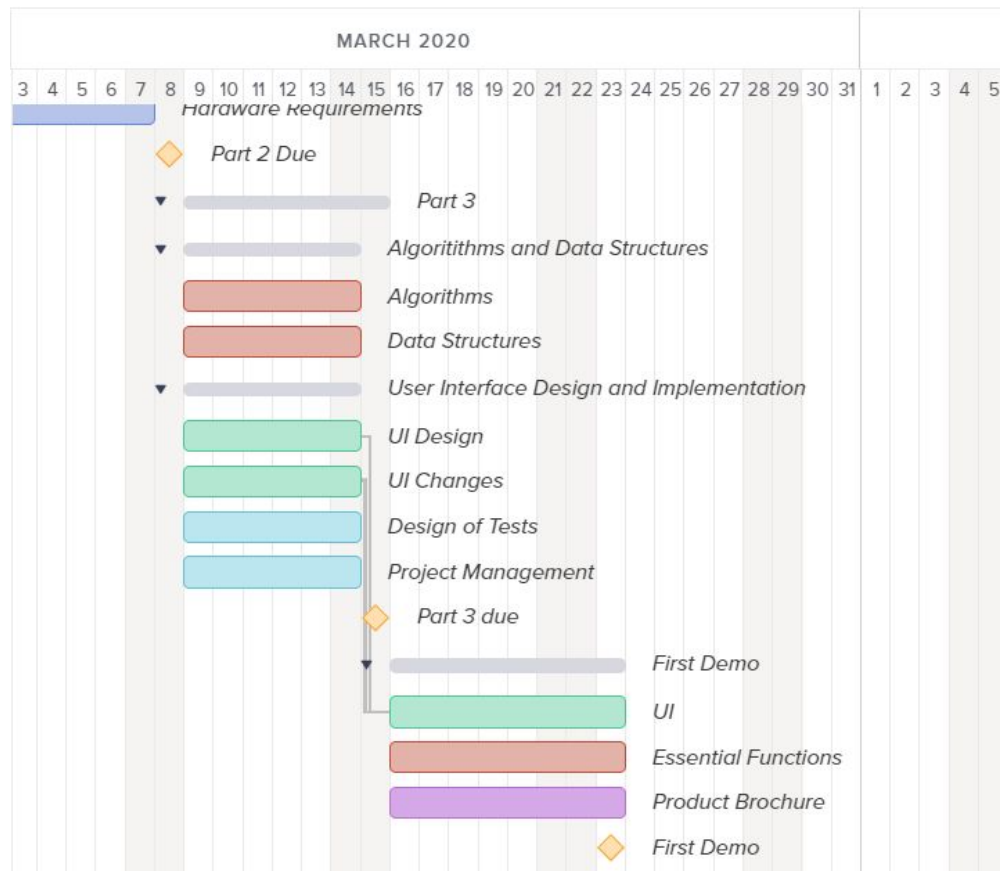


Figure 7-2

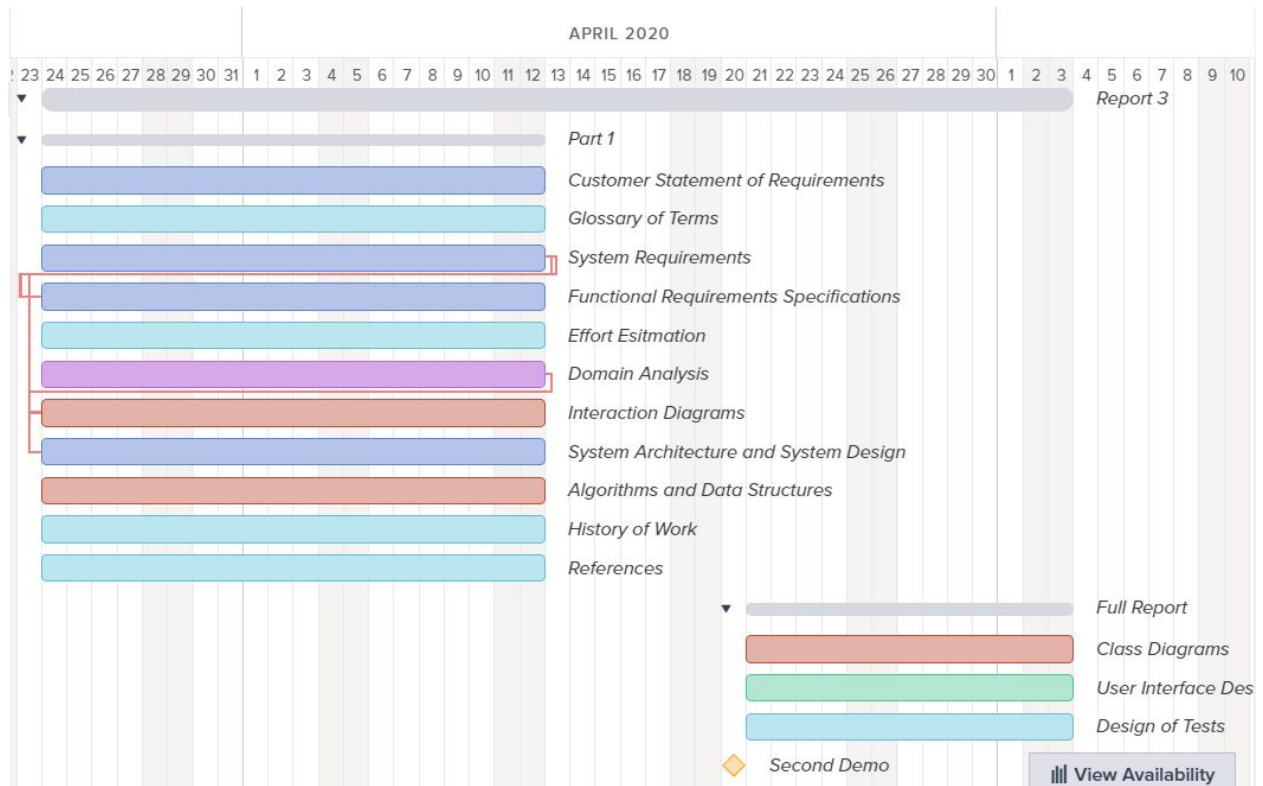


Figure 7-3

The current plan is to get the Database system and the Framework finished by March 3rd. This is to coincide with the due date of the report 2 part 1. As shown in [Figure 7-4](#) the goal for the first demo is to have a working connection between a basic GUI and our database, which will be the focus of our program. This will allow us to ensure that we follow one of the main principles of software engineering, don't repeat yourself. In [Figure 7-5](#) we have the remaining basic needs of the program. The exact functions that are going to be shown in the first demo have not yet been decided.

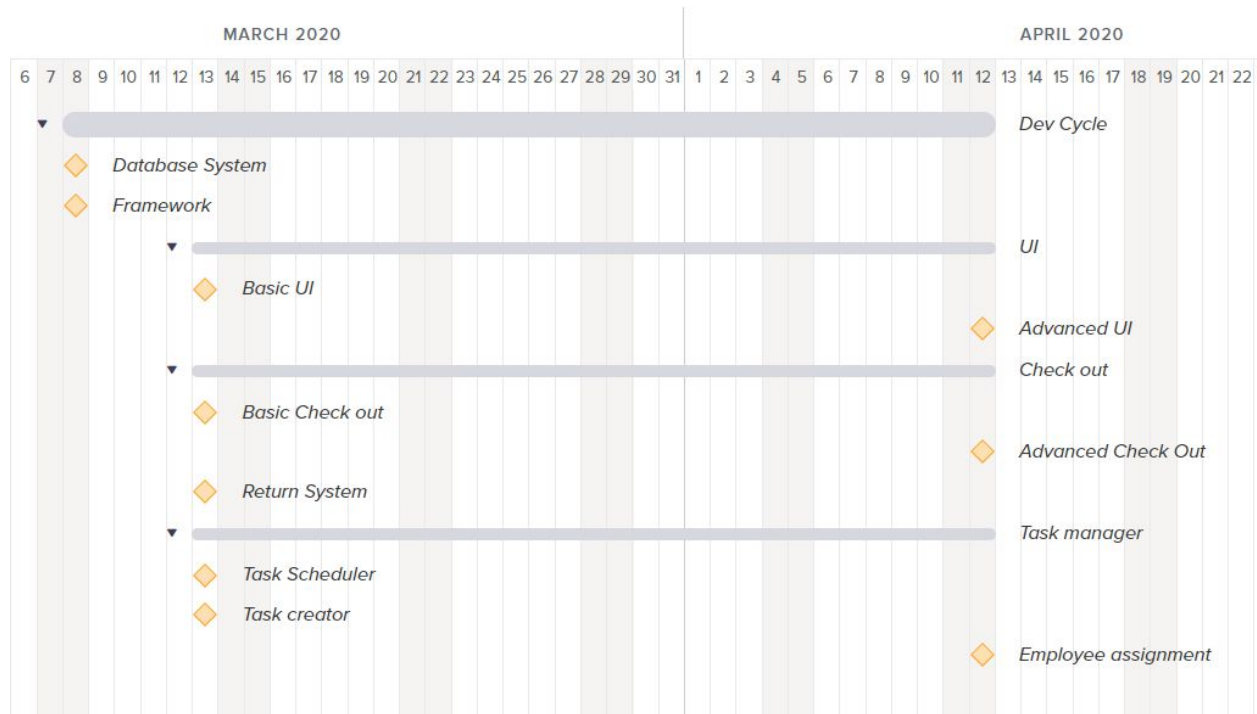


Figure 7-4

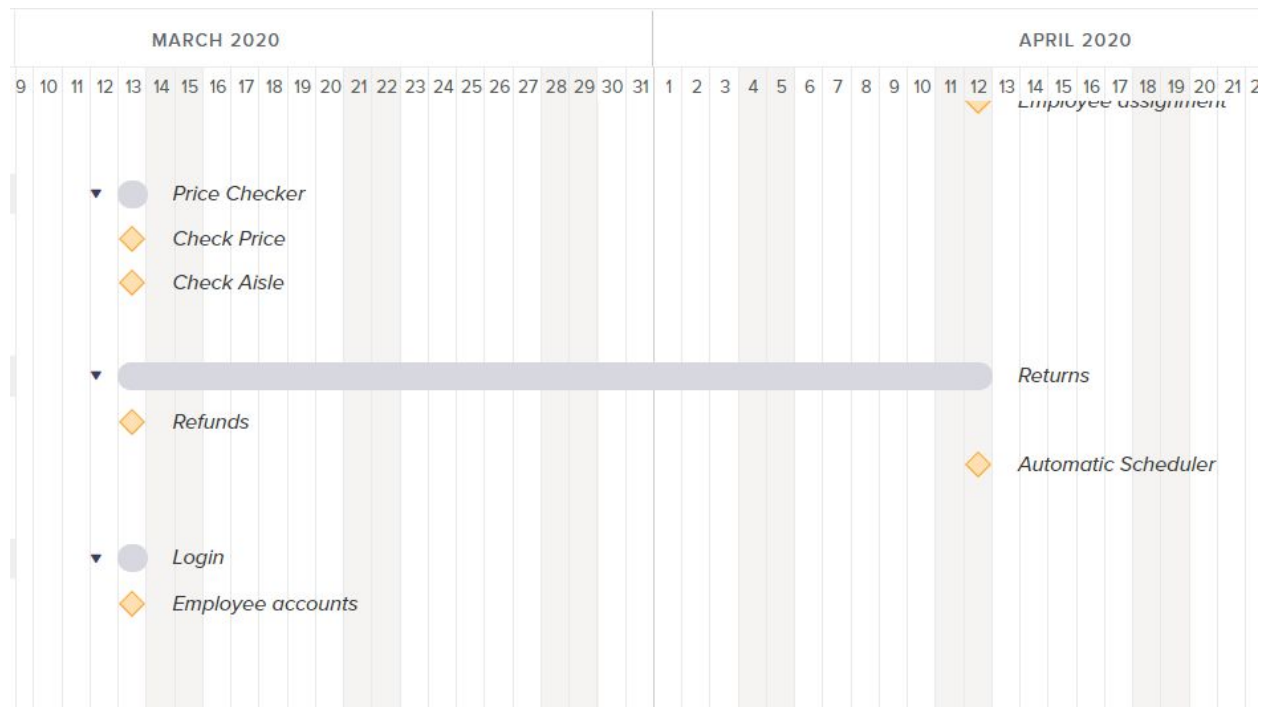


Figure 7-5



## Section 8: References

- [1] Meyersohn, N., 2020. “Another Whole Foods competitor just bit the dust.” *CNN*.  
<https://www.cnn.com/2020/02/03/business/earth-fare-grocery-stores-luckys-market/index.html>
- [2] Allen, Scotty. “Inside the RFID Stickers from a Chinese Cashier-Less Store.” *YouTube*, 21 June 2018. [www.youtube.com/watch?v=0QKrHi-G9WQ](http://www.youtube.com/watch?v=0QKrHi-G9WQ)
- [3] 2018. “Supermarket Facts” *Food Marketing Institute*.  
<https://www.fmi.org/our-research/supermarket-facts>